JPRS-CST-86-021

30 MAY 1986

China Report

SCIENCE AND TECHNOLOGY

FBIS FOREIGN BROADCAST INFORMATION SERVICE

REPRODUCED BY
NATIONAL TECHNICAL
INFORMATION SERVICE
U.S. DEPARTMENT OF COMMERCE
SPRINGFIELD, VA. 22161

5
106
A06

NOTE

JPRS publications contain information primarily from foreign newspapers, periodicals and books, but also from news agency transmissions and broadcasts. Materials from foreign-language sources are translated; those from English-language sources are transcribed or reprinted, with the original phrasing and other characteristics retained.

Headlines, editorial reports, and material enclosed in brackets [] are supplied by JPRS. Processing indicators such as [Text] or [Excerpt] in the first line of each item, or following the last line of a brief, indicate how the original information was processed. Where no processing indicator is given, the information was summarized or extracted.

Unfamiliar names rendered phonetically or transliterated are enclosed in parentheses. Words or names preceded by a question mark and enclosed in parentheses were not clear in the original but have been supplied as appropriate in context. Other unattributed parenthetical notes within the body of an item originate with the source. Times within items are as given by source.

PROCUREMENT OF PUBLICATIONS

JPRS publications may be ordered from the National Technical Information Service, Springfield, Virginia 22161. In ordering, it is recommended that the JPRS number, title, date and author, if applicable, of publication be cited.

Current JPRS publications are announced in Government Reports Announcements issued semi-monthly by the National Technical Information Service, and are listed in the Monthly Catalog of U.S. Government Publications issued by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

Correspondence pertaining to matters other than procurement may be addressed to Joint Publications Research Service, 1000 North Glebe Road, Arlington, Virginia 22201.

JPRS-CST-86-021

30 MAY 1986

CHINA REPORT

SCIENCE AND TECHNOLOGY

CONTENTS

HIGH-TECH AGREEMENTS DURING DANISH LEADER'S VISIT

Electronic Photographic Equipment Contract

Copenhagen BERLINGSKE TIDENDE in Danish 25 Mar 86 Sec. 111 p 2

[Article by Thorkild Dahl: "Eskofot Has Received an Order from China"]

[Text] Eskofot in Ballerup has, in connection with Prime Minister Poul Schluter's official visit to China, signed the first contract for the delivery of advanced graphic equipment. Eskofot has worked on the Chinese market for four years with the East Asian Company as its agent.

"The door to China has only just been opened. We are very glad, for we have made a great effort to enter the Chinese market," Director Borge Nielsen of Eskofot said to BERLINGSKE TIDENDE after signing the first contract in Peking on Monday.

Eskofot is the world market leader in graphic equipment. In 1985 the turnover was approximately 425 million kroner. Production takes place in six factories in Hjorring and one in Skagen.

Outlook for More

"We received the first order today, and at the same time the Chinese said that this year they intend to buy between 20-50 electronically operated reproduction cameras in the graphic area," Nielson said. He concluded the contract during Prime Minister Poul Schluter's official visit this week to China.

Eskofot has worked in the Chinese market for four years with the East Asian Company as its agent and has therefore had all the East Asian Company's well-deserved good reputation and knowledge to draw on.

Last fall the Chinese were sent one of the new reproduction cameras completely equipped with directions in Chinese, and Nielson says that the Chinese have tested the apparatus now for two months.

Strong competition Puts Pressure on the Price

In an ordinary sale, an apparatus costs 80,000 kroner, but Director Nielson does not hide the fact that competition in the Chinese market has made the Chinese price lower. Nielson did not want to say what the price was.

Nielson says that there is "no doubt that the order is a breakthrough for Eskofot, not just for cameras, but also for our other products such as developing machines."

In the negotiations in Peking Eskofot has now gotten its name on all relevant lists for Chinese state purchases, and Director Nielsen thinks that the firm's leading position in the world market can also rub off with a leading place in the coming Chinese market.

Know-How for Electronics Industry

Copenhagen BERLINGSKE TIDENDE in Danish 29 Mar 86 Sec. 111 p 2

[Article by Thorkild Dahl: China Seeks Products for Industry and Education"]

[Text] The opportunities for Danish firms in the Chinese market are good. If one has a product, of course, that the Chinese can use, and if the product can contribute to the development of either the industrial sector or of industry.

This is what Director Per V. Bruel from the firm of Bruel & Kjaer Export, Inc., in Naerum says. The firm makes instruments for sound measurement, whether it is to reduce noise or to make the resonance in a concert hall better.

Buying Power the Same as One Hundred Million Europeans

Director Bruel assesses the Chinese market with its one billion people to be like a West European country of 100 million people with good buying power. This corresponds to West Germany and France together.

But one should not believe that Danish firms will now just go through the door with suitcases in hand. First of all because there will certainly be something wrong in the suitcases the first time, and then because one does not speak a "language" the Chinese understand. One must learn the ropes and the customers to a certain extent. "We are interested in getting over there," Bruel said. Bruel is in China for the 37th time. The first time was in 1953, and now the firm in Naerum has two permanent employees who travel around in China.

There in the Cultural Revolution

"We consider the possibilities of development in China to be excellent. We were also here during the cultural revolution in the middle of the 1960s, and what we found out has been of use.

"It has been good for us to come here and to present our apparatus and to cultivate our connections. Most of the other West European nations stayed away, because they could not sell their products, and so there was no reason to pay the travel costs. We came, and the Chinese have been very happy about this. One must not forget, you see, that the Chinese are in a way -- well, I don't want to call them conservative, but very loyal in that they like people very much that they know and who they know have never let them down and with whom they can talk," Bruel said.

Started with a Small Turnover

Bruel & Kjaer started with a very small turnover in China, but it constantly grew. Bruel says that there is reason for continued optimism, for the Chinese have expressed a desire to expand the connections.

The turnover for Bruel & Knaer's export to China is now between 50 and 70 million kroner a year. The total firm's turnover is about a billion kroner. Bruel & Kjaer are the leaders in the Chinese market, but there is competition, especially from the Japanese.

"They are trying to live up to our standard, and the prices are half of ours. Our product is better, however, and even though it is a developing country, China still really has a need for high technology."

China is trying to come up very quickly to a considerably higher position and is jumping up many steps, for example in the electronics industry. They are trying to take great strides, and at the same time they want to get exports going to other countries, and the products the Chinese will make such as household machines, telephones, loud speakers, hi-fi equipment, automobiles -- even spare parts for aircraft --will bring them up to a level where there is the possibility of export."This means that they will have the same demand for our products as do customers in West Europe or the U.S.," Bruel says.

The requirements for export products to China are that they can be repaired and in general last a long time. The quality must be tops, for no one in Denmark can sell cheaply to the Chinese, and Bruel says that "no matter what we produce, it is more expensive than any other product, and therefore quality or innovation must be built in." One must not forget to see things from a Chinese point of view. They must have something for their money, and if it is to be good business, they must have some use for the product as well. They must have a product that will fulfill the demand for quality, and if they order something else, then one must most certainly agree and get it done," Bruel said.

To a certain extent, the opening of China means that the authorities in the provinces will be able to make decisions themselves.

"We are naturally dependent upon the purchasing organizations, and it is important to have very good relations with them -- no doubt about this. But we now have the possibility of getting through to the real customers, and this is a great advantage for us that we can explain the things we have and set up

seminars and small exhibits and visit firms and educational institutions directly. To a great extent we can also bring Chinese specialists home to us in Naerum and give them a course in a certain discipline," Bruel said.

Soon the director will celebrate his fortieth trip to China, and after all these trips he says that the same mentality dominates now as before among the Chinese, even though the changes have been great. "They like people they have known a long time and they tend to make agreements with them. One does not have to have anything in writing, but one has it, naturally. We have never gotten into an argument or had any problem at all with a contract. Honesty is necessary for trade with the Chinese, and Danish firms that play their cards in this way will fit in splendidly in the Chinese market," Bruel said.

Service Net will be Expanded

If one does something wrong, orally or in writing -- that is, something that is not in agreement with the truth -- then one had better go home at once and stay far away and never come back. This is Bruel's good advice.

Bruel & Kjaer Export is working hard to expand the service net in China, and two centers are already on the drawing board.

"We now have two service centers in China, one in Peking and one in Shanghai. It is very important, you see, when one is to trade here, to be able to see to it that the products one sells last a long time. High technology products require a type of service that is perhaps a little difficult to build up, because one must have the personnel who know the products in and out and who have the possibility to repair them. There must also be enough spare parts. We were the first firm in the world to set up a service center in China manned solely by Chinese engineers," Bruel said.

QIAN XUESEN ON SUPER COMPUTERS, INTELLIGENT MACHINES

Shanghai ZIRAN ZAZHI [NATURE JOURNAL] in Chinese Vol 8 No 1, Jan 85 pp 3-9

[Text of a speech given by Qian Xuesen [6929 1331 2773] at the "National Defense Science and Technology Committee Conference of Fifth Generation Computer Specialists" held on 3 August 1984: "On the Question of 'Fifth Generation Computers'"*]

[Text] These days everyone is talking about the so-called fifth generation computer, about which there is much discussion both here and abroad. I believe that the central question here to resolve is: What do we mean by "the fifth generation computer"? I discuss below what I know about this question, as well as some preliminary opinions on developing work in this aspect. I like to discuss this subject from the angle of 'science of thinking'. My thinking might not be along the same lines as you, so comrades where I am off, please comment.

Supercomputers

I think that I will start with the easiest and save the hardest for last. I will begin with the fourth generation computer. That is, from the point of view of how to fully develop the function of the current supercomputers(1) (like the "Galaxy" brand computer) the goal is to develop the scientific applications of the computers.

Current supercomputers have already broken up von Neumann structures and have introduced parallel operations. However, there are still questions about how to make full use of that. Which is to say that we still do not really know how to use this kind of operational structure, so the potential

* This was read on 3 Aug 84 during the "Fifth Generation Computers Symposium of the National Defense Science and Technology and Industry Commission."

(1) We assume here that computers with a large number of operations are in the range of 10 million per second. Medium size minicomputers are less than 1 million operations per second. Microcomputers are less than that. What we mean here by supercomputers are computers that have significantly greater than a large number of operations. We could call them megacomputers.

of computers has not been fully realized. With similar foreign computers, like the Cray-1 and the Cyber-205, users still do not know how to use them, and full use cannot be made of most of them. This is a general problem. I believe that this is a question of four aspects.

1. Concerning non-linear partial differential equations

We know that when partial differential equations are linear the nature of their solutions is clear, but there are still many problems about non-linear equations that have not yet been solved. I have had a little personal experience in this regard. We had already discovered in the 1940s that because the differential equations in aerodynamics for non-viscous gases are non-linear, solutions do not exist for these differential equations under all conditions. Sometimes, although the speed of an object in motion is subsonic, when the speed increases to a certain Mach number there do not appear to be solutions for ideal gases.[1] This is because we are unable to solve for them and what we have is a guess. As far as I know this theoretical problem has not been solved even today. The solution of non-linear partial differential equations is rather more complicated. In our current actual practice many engineering and technical problems involve problems with non-linear partial differential equations. I discuss below three problems along these lines.

Regarding the finite element solution method for non-linear equations. We are aware that currently when computers solve partial differential equations they use the finite element method. How ought we to use the finite element method of analysis to solve non-linear partial differential equations? This is a question we are just now working on. For example, I happen to know that Professor Zhang Xianglin [1728 4161 7792] of the Beijing Industrial University is studying this question.[2,3] This is an important problem that should be well researched.

Regarding the method of multiple-order perturbation calculation. That is, if we do not use the finite element calculation method, but use the multiple-order perturbation calculation method to solve non-linear partial differential equations, we use symbolic manipulation in the computer and not numeric calculation. This method is very valuable in aerodynamics because the solutions that are obtained are not for a particular [lai liu] Mach number, that is, solution for an M number, but there are also solutions for other M numbers as well. In this aspect is also included the singular perturbation method. That is, when a small parameter emerges from the perturbation method it is combined with the highest order partial differential equation, which is rather troublesome. There are people in China who are also studying this question.[4] Work on this aspect has already been done mathematically, and the problem is how to use this method in the computer using symbolic manipulation rather than numeric. So far, no one is studying this problem in China. I once exchanged opinions on this with Comrade Zhang Hanxin [1728 3211 0207] of the China Aerodynamics Research and Development Center. He feels that work ought to be done on this, but that the problem is quite large and needs a plan and an organizational force before it can be taken up.

Regarding the nature of partial differential equations. The solution of linear partial differential equations is rather regular and is not likely to produce trouble, so we need not be concerned about these. In the solution of non-linear partial differential equations, as in the example above about study of the motion of non-viscous ideal gases, there is often trouble. Under these conditions, if the nature of the solution is not understood beforehand, when it is then solved for, the result will very possibly not be true. Therefore, if we have some sort of understanding of the nature of the solution of the non-linear partial differential equation before calculation, that is, under what conditions there might be a special expression, we can be on guard and can take measures against that during calculation. This problem involves the mathematical fields of differential geometry, differential topology, and differentiable manifolds. For example, if we want to solve the problem I have mentioned about the highly critical M number, then we must begin with a study of the nature of the solution. It would appear that there is already mathematical preparation on this question. I do know that Comrade Zhou Yulin [0719 3022 7792] of the Ministry of Nuclear Industry had been concerned with this problem in practical work[5]. It is my feeling that a method to respond to this question should not only be confirmed theoretically, but should also be answered concretely. This is to say that we want also to be able to use the computer to analyze the nature of solutions of non-linear differential equations, and we want to be able to respond with the computer, just as we use the computer to prove four-color theorems, because it is just too much trouble to do manually.

Speaking in general and proceeding from the view of computer math or computer science, we need to use our current supercomputers well, especially those supercomputers with parallel operations. This is to say that we want to learn to use our supercomputers intelligently, that it will only be satisfactory if there is an integration of people and machines; otherwise, we will not be able to develop fully the potential of the computer and could even obtain erroneous results. Therefore, we want to first solve the problems with parallel operations. In addition, we want to research three questions: (1) non-linear finite element analysis; (2) multi-order perturbation methods; and (3) regarding the nature of the solutions of non-linear partial differential equations, that is, whether or not a solution exists, we want to predict when special conditions will emerge, and we want to use the computer for predictability analysis. Of course, we must also study questions like machine software and operating software.

In the field of computer mathematics, I believe that these questions must be solved. If this is so, do we want to hold a conference of specialists to study these questions? This would mobilize many more scientific and technical people to do this work than are currently active. We need the help of the mathematicians. This would of course also open up the mathematical sciences and stimulate the development of mathematical science. On this question, Comrade Gu Chaohao [6253 6389 6275] of Fudan University in Shanghai has written some articles[6]. He points out that the development of mathematical science is closely related to computers, and I agree with this. This question ought now to elicit our respect. In the past we were busy with creating machines and felt that it would always be useful to

produce machines, but we did not greatly respect how to use them. Therefore, although the machines have been produced we still do not know how to use them very well. This question must be on our agenda, because only then can we fully develop the function of the supercomputer.

2. The Significance of the Supercomputer in the New Technology Revolution

What we have discussed so far is the question of how to develop fully the function of existing computers. Now I want to talk about the fact that current computers have simply broken up the von Neumann pipeline, single line operations by adding parallel operations. In further developments there are even greater supercomputers. The significance of this development is very great. There are three companies in the United States that build supercomputers, those companies being all very small in scale: one is the Cray Research Company, the computer they produce being called the Cray-1, and currently working on the Cray-XMP; another is CDC, original makers of the Cyber-205, from which company has come a division, ETA Systems Company, which does only supercomputers; the third company is called the Denelcor Company, which produces the HEP-1. All in all, the operational speed of these machines is tens of millions of floating point operations per second (M flops), which is the current level that has been reached by current supercomputers. They are used in engineering technology, as for calculations in aerodynamics to substitute for wind tunnel experiments, for turbine blade analysis to substitute for turbine blade experiments, etc. To manage these tasks current operational speed is still rather low. For the two kinds of calculations mentioned above to be effective the time of each result must not exceed one second. This shows that high technology and hyper-technology want to use calculation and analysis by supercomputers to take the place of complicated and expensive experimental problems. Two issues of the U.S. AVIATION WEEKLY, 1984.5.28 and 1984.6.4, had articles specifically on this question. The analysis by that periodical considered that there is not a large market for supercomputers because at present the operational speed of these machines does not suit the problems they are to solve, that is, the speed is still too slow. For that reason as large a computer company as IBM does not make supercomputers, feeling that the profit is too small. The three companies mentioned above are all rather small. As of now, only 65 Cray-1 units have been sold, ETA Systems Company has sold only 25 Cyber-205s, and Denelcor Company has sold only four of its HEP-1. This means that only about 100 American supercomputers have been sold. They feel that unless there is a big technological breakthrough, probably only about 400 will have been sold by 1990.

To make that breakthrough, current operational speeds must be greatly increased. This is perhaps one way to understand the fifth-generation computer, that is, that it is just breaking up von Neumann structure with parallel operations, which understanding would be in actuality a further development of the fourth generation computer. If this kind of computer is to truly substitute for the very expensive experiments of engineering technology, then its operational speed must not be just tens of M flops but must be thousands of M flops. That would be to increase current operational speed from 10 to 100 times. We may accept this understanding

of the fifth generation computer because using this kind of computer in engineering technology would always save time and money in doing large scale experiments.

Understanding the fifth generation computer in this way, although it is simply as a development of the fourth generation there are still many problems. Besides the problems with computer math discussed above there are also hardware problems. It is our goal that the speed of operations be from tens to a hundred times quicker than current computers, but from the point of view of the development of current semiconductor devices, there are probably limits to further increases in fundamental frequencies. GaAs devices might be used but the increase is only a few times and cannot increase tens or hundreds of times. Although this is true we cannot abandon our efforts in this direction and should continue to work with this. However, it appears that a more workable method is to increase the number of parallel arithmetic units. Current supercomputers have from 2 to 4 parallel arithmetic units, but Americans believe that in the late 1980's this will be increased to from 8 to 16, and by the late 1990's could go further to 60 or more. But as it is, we still do not really know how to use the current machines with 2 to 4 parallel arithmetic units. If we are to increase the number of those units in the future this mathematical problem will be even greater. But we must work in this way because if we do not, operational speed will not increase to the level required by actual applications.

3. Supercomputer Design Studies

There is also a specially designed computer outfitted with this kind of arithmetic unit called a data flow computer, which is suited for matrix calculations. I have recently seen an advertisement by the American FPS Company (Flating Point Systems), which has a machine called the FPS-164/MAX. This machine is truly specialized for matrix calculations. It is said that this machine takes only about one second to find the factors in a 1000 X 1000 matrix. It would take about two hours to multiply two 10000 X 10000 matrices, which is already at a rate of 300 M flops. They advertise that although this machine operates at 300 M flops it sells for only $1 million, which is quite inexpensive. But when you read on it is clear that it is specialized for one particular task, which is matrix calculation. If we then look at the problem in this way we feel that there is an alternate path, which is to use the matrix calculations for optical lenses. Everyone knows these and they have been used before. But previously optical lenses have been used to do model calculations, and since accuracy has not been great that is a limitation. But I have recently read an article in an American publication [7] that talks about numeric matrix arithmetic units for optical lenses, which are already beginning to be used. Then, as this technology develops in the future that would combine optical lens matrix calculations and their ultra high speed with the accuracy of numeric calculation, and this I feel is also a direction. Therefore, within parallel processing great numbers of matrix calculations will be encountered. The calculation of matrices can use special matrix arithmetic units. From the view of much further development, that is, optical lens numeric matrix arithmetic units, speed will be even faster.

I want to mention further that at the 1977 meeting at the Friendship Hotel in Beijing to discuss the current Galaxy machines I brought up a few questions. At that time we were all thinking about making the Galaxy computer and were busy working on the machine. Therefore, there was no time to consider long term issues, and I probably raised those issues too early, but I feel that we ought to consider them now. We must realize: there is a very great difference between the development of current computers and the development of computers in the 1950's. Speaking vividly, we might say that, before, components were very expensive and wire was cheap. Now? It is just the opposite: components are cheap and "wire" is expensive. How could wire be expensive? Because as wire gets longer operation speed cannot keep up, so wire becomes a troublesome thing. I remember seeing once some reference material that was not talking about our current Galaxy supercomputer, but rather mainframe computers in general. It said that how much of the cost, after all, did the components account for? A very small part because the greater part of the cost was spent for "wire," that is, in the construction of the machine. This then raises a question well worth our consideration. Is our current design thinking still along the lines of the thinking of the past, where components were expensive and wire cheap? Or is it just the opposite, where components are cheap and "wire" expensive. Is there an error in our guiding thought in regards to handling construction design?

This then brings up another point, the construction of machines, where geometric distribution is extremely important. Not long ago, I also saw something [8] which said that because of the developments in large-scale integrated circuits there may be many components on a board. In this way, the construction of machines has changed, and the path of signal transfer is shortened, consequently quickening its speed. Then, this is to say that when designing machines we also want to consider topological construction and geometric distribution. This is a very significant problem and is a fundamental restructuring, and is well worth our study.

I remember that in 1977 I brought up another question, which was, since components are now cheaper, could more components be used to improve the speed of computers, and would that be possible? In the past it was because we wanted to save on components. When we have designed logical and arithmetic constructs, now that components are cheaper could we change? I know that later Comrade Luo Peilin [5012 3099 7207] of the Ministry of Electronics Industry did some work in this regard that proved this was possible. By using more components speed will be increased. I have heard that Comrade Wang Shoudang [3769 1343 8093] of the Institute of Semiconductors of the Chinese Academy of Sciences has also paid much attention to this problem. I did not propose questions like these in complete detail. They are questions that belong to our further advances in computer development and will further improve operational ability. In this regard, we ought to break open some of our old conventions.

The above are ideas I presented in 1977, and now that we have produced our first generation of supercomputer we must consider how we are to improve speed on the basis of this first Chinese supercomputer. We must discuss

these questions on scientific principles and not blindly rush ahead. If we ought to research something, ought we then to call a special conference? We would invite everyone and fully discuss questions about the machine itself.

4. The Future of Supercomputers is Further Development

At present, will we be likely to raise this kind of question: you say that we should do all we can to improve the operational speed of computers, you say that current computers are too slow, that they should be from tens to 100 times faster. Then in the future will you say that 100 times is still not enough and that it must go faster? Is there any end to this?

What is the significance of this idea? One is as I have just discussed, that in high technology and in hyper-technology it has significance. Some of you sitting here today are involved with nuclear technology and it is of great significance to nuclear technology. For nuclear explosion technology to further develop we need even faster calculation speed.

Aside from this, this idea gets into much broader fields, that is the field of natural science research. You all know that computers are already being used in natural science research. In, for example, quantum chemistry, to use the computer for analysis of atomic structure is what is called computer chemistry. With further developments some say that it will not be necessary to do experiments for many chemical reactions. Rather, computers will do it. Or, for those things that are difficult to produce in experiments, the computer can do it. With further developments it will get so that the computer can be used to design chemical molecules with particular characteristics. To do this, computer power must of course be very large. For astronomical studies. Everyone knows that the processes are generally very slow. If a person wanted to wait until things change, even several generations will not see it. What can be done? These processes can be modeled in the computer to see whether or not they are correct. If they can be modeled then the theory will have a basis. This method has already been used in space science. For example, in the Big Bang theory, after the explosion how did the unevenly distributed universe that we have now come into being, where in some areas there are more stars and star systems and in some places fewer? How could this have happened? Etc., etc. This kind of question can be modeled on the computer, and the result of the model is very clear.

At present, other aspects of natural science research, as in ecology, are using the computer.

Although these examples explain that computers can forward the development of natural science, it cannot be said that the development of natural science depends upon the computer. There has been a recent dramatic example [9]. People are trying very hard to understand a current fundamental question in physics, the problem of baryons. As you all perhaps know, physicists have worked out a theoretical framework, which is the so-called quantum color [she 5331] dynamics. Unfortunately, quantum color dynamics is extremely complicated, even more complicated than quantum electrodynamics.

Therefore, although we have the theoretical framework, how can we use the theoretical framework to obtain some results. The necessary computing power is incredibly large, and there is no way to compute it. In 1974, in the early period of quantum color dynamics, the American scientist, Kenneth Wilson (winner of a 1982 Nobel Prize), suggested that the four-dimensional space grid method be used to solve problems with the strong-interaction theory. That, then, is to make finite elements, form the successive fields into a grid, and then you may solve by the Monte Carlo method.

Not much note was taken of this suggestion of his among physicists of the time. Two years ago other methods had definitely not worked. I remember that at that time, at a conference on elementary particles held in Guangzhou, we discussed excitedly the so-called grid method, and then applied it last year. But we discovered that this method requires great computing ability and general computers will not do. So, physicists then became computer advocates and searched everywhere for large capacity computers. Well, we now have some preliminary results. We feel that there is much hope for the use of Wilson's suggested four-dimensional space grid method. This can explain some special problems in quantum color dynamics, as for example the confinement of quarks and even theories on aspects of elementary particles, such as can we predictably calculate the mass of all kinds of baryons? It is now considered possible, but use of the Cray-1 computer is still too slow. At present, theoretical physicists in America have joined to study this problem and have asked the government to support a plan for a computer engine, which would be used to solve problems in quantum color dynamics. They have proposed that the computer is highly crucial to solving fundamental problems of physics. Without the computer there is no way to go on.

Comrade Cheng Kaijia [4453 7030 3946] of the Scientific and Technological Committee of the National Defense Science, Technology, and Industry Commission feels that we have already reached the point where we cannot go further in creating a high-energy accelerator, the problem being that the expenditure would be too great. If we are to reach even higher capabilities, one way to do that would be through the computer. I feel that this is very significant, that is, not only for engineering technology but also for natural science and basic science to develop further, very-large-scale computers are a key item. Thus, in this way we can join engineering technology, natural science, and scinetific technology together. The conclusion is that unless there is some breakthrough in principle on the basis of the fourth generation computer, that is, where the capabilities of the computer are raised another tens of times, hundreds of times, or even several hundreds of times, this will have a very great significance for engineering technology, natural science, and basic science. It might even be critical.

We will want to earnestly deal with this question. If the situation is as I have said, then we ought to diligently study questions of this nature together with questions from computer mathematics, and from machine and other areas to lay the foundation for basic principles and policies of our

country toward future supercomputers. This work is obviously that of a national scale. We already have supercomputers. Can we consider the so-called fifth generation computer to be the second generation of supercomputer, a further development of the fourth generation computer? This is one answer to the question of what is the fifth generation computer. What is the fifth generation computer? If it is the second generation of supercomputer that does not include work with other computers, but only deals with supercomputers and relevant work on them.

Artificial Intelligence

There is another answer to the fifth generation computer, which is what we now want to discuss. This other answer is what Edward A. Feigenbaum and American female writer, Pamela McCorduck, have pointed out in their book, "The Fifth Generation: the Challenge to the World by Artificial Intelligence and Japanese Computers." They feel that what the Japanese have spoken of as the fifth generation computer is an important computer revolution. The function and concept of this kind of computer are both different from the earlier fourth generation. Later, the American BUSINESS WEEK said, if the Japanese machines are produced they will create a heavy burden on knowledge, which will transfer from a person's brain to the machine. Of course, this sentence is not quite accurate. Generally speaking, this kind of concept of the fifth generation computer is completely different from the concept of the fifth generation computer that I have described above.

1. New Content: Thinking in Terms of Images (Direct Impressions)

About what new content were the Japanese thinking when they conceived of the fifth generation computer? In general they wanted to add graphics information processing to computers, for it to be able to recognize images. There would also be knowledge information processing systems, expert systems, and knowledge bases. Finally, they would organize these with the machine's logical operations to form a system. Then, if we look at this question from the point of view of the science of thinking, to include graphics processing systems, knowledge information processing systems, and expert systems, all with their characteristics, then these things would truly make breakthroughs in simple logical thinking, that is, from the constraints of abstract thinking, which already includes factors of thinking in terms of images (direct impressions). It is my belief that from the view of the science of thinking, thinking in terms of images (direct impressions) is not the same as logical thinking. For it to make a breakthrough in logical thinking and abstract thinking would be a very great breakthrough. The first point in earlier computers, with their von Neumann structure, was logical operations. Then, with the fourth generation computers and on to the future second generation of supercomputers we talked about, parallel operations are fully developed. But this has not been a breakthrough in the fundamental principles of logical thinking and logical reasoning. At present, the Japanese are talking about graphics processing systems, knowledge information processing systems, and expert systems. These things are not the same, they are not limited to logical

reasoning, but rather are broader than that. Broad in what aspects? That is, by taking into account the factor of human experience, which question I will now take up.

2. Pattern Recognition

A topic that has been studied fervently over the last 10-odd years is what is called pattern recognition. It refers to recognizing an image, as for example being able to read. The ability to read is a tremendous thing, and people can read both sloppy handwriting and calligraphic flourishes. How about when using a computer? It does not work then. For example, libraries abroad have machines that can read that are used by the blind. They can recognize words that are printed and read them, and the blind can read the same books we do by using this machine. But they can only read printed forms, and handwritten forms or printed forms other than those specified cannot be read. There is also phonetic recognition. If two people are talking, even if they have different accents and make grammatical mistakes, or even if other things get into it, it makes no difference, each can understand the other. But when speaking with a machine simple language, like commands, is all right, but when it comes to conversation machines cannot understand it, and there is just nonsense. Pattern recognition has been going on abroad for more than 10 years, so where are the problems? I feel that the original studies of pattern recognition were compltely based on logical reasoning, that is, the method of abstract thinking. In actuality, pattern recognition in people is not just logical reasoning, but is also thinking in terms of images. In this there are empirical factors, which means that people know things from experience. They know what is possible and what is impossible, and in this way greatly simplify the reasoning process. Of course, individual abilities are within a particular scope and anything within that experiential basis is alright, but once outside one's scope of experience is exceeded it is no longer so. Look at the recognition of words. I often receive letters from others, most of which I can read, but some of the words I cannot make out. Some of the writing of younger people is especially strange, which stumps me for I cannot make it out in any way. Reading from beginning to end I still do not know what they are talking about. That has exceeded my scope of experience and there is nothing that I can do. Or, look at people listening to speech. It takes a long time before children can understand by listening. At first they can only understand very simple speech and cannot understand that which is more complicated, for this is a process of the accumulation of experience.

I have heard that specialists studying pattern recognition abroad run into a dead end after several years of effort and cannot go on with pattern recognition. At present there are those of that group who are very interested in expert systems. Well then, what are expert systems? Expert systems include experiential factors. What is more, there has recently been a breakthrough in linguistics research in which certain concepts have been proposed that say that language is based upon knowledge. I have wondered, what is a knowledge base? It is just people's experiences. These two pieces of information explain the importance of thinking in

terms of images, because within image thinking actual experience is a factor. Besides the examples given above I could also cite the example of appreciating works of art: without the process of learning and comprehending you cannot appreciate art. What I mean by learning is to have an experiential base, and without that, thinking in terms of images is empty. It was Lu Xun who said years ago that people of different experiences cannot have the same feelings of beauty. Therefore, the so-called thinking in terms of images (direct impressions) is to have an experiential basis, and is not completely a process of reasoning.

At the moment, when the Japanese speak of including these things in the fifth generation computer, that is to include thinking in terms of images (direct impression). Moreover, it now looks as if it breaks up pure logical reasoning to include experiential factors. Of course, the so-called experiential factors are certainly not just experience. Experience must be combined with reasoning before it can be of use. I can give an example of this. There is a foreign psychologist who saw a peasant in a remote, dry area. This peasant had been planting there for many years, even all his life. In normal harvest years under dry and warm conditions he knew very clearly how much he would harvest in a year from one mu of land. That is, whether it would be 100 or 80 jin. He knew this from experience. When the psychologist asked him about this he was very positive in his answer. The psychologist then asked, "There is an area in a foreign country that is as dry as this and about as warm. If you were to plant there, how much would you harvest in a year?" The peasant replied, "I don't know. I've never planted anything there. There's no way I can answer." When asked again, he always answered, "I don't know." This shows that this peasant had the experience of planting his own area but that he could not reason, he could not think in terms of images. This point has been demonstrated in an article by Comrade Ma Xiwen [7456 1585 2429] [10].

3. Expert Systems

What do we mean by breakthroughs in the framework of logical thinking? Simply, to introduce the factor of experience, and this is extremely important. What are called expert systems are the opinions of an expert, they can only tell that it would be correct to do something in a certain way. If you ask why, it cannot explain clearly. There are many cases like this. When I was once teaching at the Chinese University of Science and Technology some of the students asked me how I was able to calculate integral problems so quickly, what tricks did I have? I replied that there is no way to say, it is just from much practice. Being unable to explain the reasons, that is what we call the factor of experience. We said before, that this kind of experience must be combined with reasoning before it can be of use.

Well, then, what is it that can combine the experience factor with logical reasoning? This is a great question in research in the science of thinking. At present, the greatest question in the science of thinking is this thinking in terms of images. It seems to never be explained clearly. I feel inspired now that this is being linked to some developments in what

the Japanese are calling the fifth generation computer. What is this thinking in terms of images? It is that having surmounted simple logic to then combine experience and reasoning, and the outstanding example is just this expert system. Expert systems are where an expert, based on his experience, having 1,2,3 will then have 9. You ask, how can you have 1,2,3 and then 9? It says that this is derived from experience. All you can do is to go along with it, for as long as it is within the scope of expert experience it is then correct.

There is an old doctor of Chinese medicine in Beijing, a liver specialist, Doctor Guan Youbo [7070 1635 3134]. His experience has already reached the status of expert system, so they did a test. First, Doctor Guan saw a patient, then the computer saw the same patient. The prescription written by the computer was given to Doctor Guan. He said it was fine, that that was the way it should be. Any errors were probably minor. That is to say that this expert system was a success.

We often say that human talent is rare, and it is even harder finding someone to appreciate it. But I say that it is not difficult for people who truly understand things to recognize talent. If I were to do this sort of thing, after talking with someone for 15 minutes I would know the extent of that person. There would be no need to expend a great deal of effort, nor would a test be needed. It would be clear after a few questions. That is to say, that within my field I am an expert at recognizing talent. Outside of it I would not be. These kinds of things can be learned by computers. What are called expert systems abroad are just these sorts of things.

I saw an article recently [11] that talked about modal logic, which was very enlightening to me. As I understand it, modal logic is something of an expansion of what we normally call logic. We insert other factors and can thus experience and make judgments. If this kind of knowledge is correct, then I feel there is hope. That is, expert systems seemed so mysterious, but now they are not. They are just the results of people's actual experience plus reasoning. If this question has been resolved, then that will have settled the expansion of use of the computer to the scope of a model of human thinking, its extension to thinking in terms of images.

There is a field within the study of thought called inspired thinking. As for inspired thinking, this is in actuality the human subconscious, which assimilates other knowledge stored in the brain to handle problems. We ourselves exist in consciousness, in a conscious state, but do not realize that at the same time the subconscious is working quietly. Suddenly, after we have achieved some result, our subconscious tells our consciousness that the problem has been resolved, which is what we call inspiration. However, from the point of view of the combination of experience and reasoning we have just talked about, there is nothing mysterious about this. It is saying, I am not in an expert system, but am in many expert systems, or that I have searched for my result within an even greater knowledge base. What is important is how experience and reasoning are combined.

It is my belief that if we were to understand the fifth generation computer the way that the Japanese have said, then that would be a basic problem. This would just be breaking up two lines of the von Neumann structure. Not just breaking up the line of "pipeline single operations," but would also destroy the line of logical operations and reasoning operations, for thinking by images. To add in the factor of personal experience, that is a great problem, one that can certainly not be solved in one or two days. It is not a question of chasing after some high speed but of being unable to clarify the entire machine structure. If machine structure is not clear then we cannot make machines.

4. Knowledge Bases, Knowledge Engineering

The Japanese concept also includes a knowledge base. A knowledge base is various individual information system. Human knowledge can now be stored by all sorts of methods and then retrieved and accessed, which is tremendous. Before, when we studied something we depended on our minds to remember it, and if we could not remember the content we had to at least remember a clue with which to search for it. If you have absolutely no idea then it matters not at all to you that that knowledge exists. Now we have a knowledge base and an information system, which are not the same. All kinds of knowledge may be stored and then under your direction will be put through to you for your use. This is a problem we discussed years ago, and is truly tremendous. In the past we scholars depended on our memories all our lives, and even when our hair is white, there we are still gnawing on books! This is called "with hoary head exhausting the classics" [meaning an aged person can still learn]. When you have a lot of books it's "making the ox sweat and filling the space" [meaning in abundance]. This is to describe a great deal of knowledge, but the ability of people to assimilate this knowledge is limited. But it is not that way now. With information retrieval systems and information transfer systems you can look up any information from any location.

To combine graphics and information processing systems, knowledge information processing systems, and expert systems, the future of that system is enough to inspire people. It will be as if a person's brain can expand to that large a scale in a moment. Not only will there be my own experience, but I will be able to absorb the experience of others as well. Not only will there be the things that I can remember myself, but things that I cannot remember will all be entered into this system. We cannot know how much this will increase a person's intelligence. I believe that this is what is called knowledge engineering, where the application of knowledge forms an applications engineering technology. I feel that this kind of concept is very important and it is not suitable to call this concept the fifth generation computer or the sixth generation computer, because it is not a computer but is an intelligent machine. Therefore, I suggest that we not confuse these things. Let's just call it the first generation intelligent machine. This kind of work is of course not the same as other work in artificial intelligence, as in man-machine, etc., aspects.

Thus, we have presented two concepts. That which is called the fifth generation computer divides into two branches. One is the second generation super computer and one is the first generation intelligent machine. These are two different concepts.

5. Studies in Intelligent Machines

If we consider the first generation intelligent machine concept of which I have just spoken, then we should further study the following problems: the first is knowledge base problems. There are some units in China studying knowledge base problems, as for example the Information Resources Institute of the National Defense Scientific and Technological Commission, which is doing preliminary work. But I feel that there are still many problems here, as for example how can knowledge bases be linked together into network searching, which is still a problem in this country. There is also the problem of entry of Chinese characters, about which there is much debate and many different methods. This requires unification, for if it continues to go on in this confused way there will be future delays, and none will be mutually applicable. I feel that this problem should not be put off any longer, but should be decided by the government. If we want to connect up intelligent machines with knowledge bases in the future the requirements for search speed are very high, and our current search speed is too low and not suitable. To scan through all of human knowledge to bring up what I want will require a search speed much higher than that we have now. We will just say this about knowledge base problems for now because in the future we will have a conference of experts to discuss it. The second problem is recognition systems for writing and speech, which we should earnestly pursue. The third problem is expert systems, which is the fundamental thing for creating an intelligent machine.

These two questions we have already looked at, so I will not say more here. Of course, the central question in making the intelligent machine is how to combine the parts of which we have spoken above with the computer's logic and reasoning operations to form a complete system. An expert system, for example, is not one expert, but is the pooling of the collective experiences of many experts, so applications questions still need to be studied. So we have here at least four kinds of problems: knowledge base problems, writing and language problems, and expert systems problems, the last of which is a problem of how to form problems in these aspects into one system, one unified system. This problem is of course a great one, so there are those who have said that the Japanese plan cannot necessarily be realized in 10 years. But I believe that this is a big matter. If we say that the appearance of the computer was a technological revolution, then the intelligent machine will also be a technological revolution. Therefore, we want to be the first, and having seen its significance must be certain to produce this intelligent machine, which would be a tremendous event. But, secondly, we must not begin work hastily and make mistakes from haste.

Some Views on Future Work

We already said above that the appearance of the intelligent machine will be a technological revolution. I still feel that only the intelligent machine,

or call it the intelligent machine system, will be what is now called the information society, or in my own words, the central problem of the fifth industrial revolution. Because we may not have this machine, in the information society the amount of work we have to do will be so much that we will be at the stage where we will have no way to do anything. Our leadership policies all need a huge brain trust, but it probably will not work if in the future it will increase to the degree where sole reliance on human effort will make it difficult to work without an intelligent machine to take up some of the work. Well then, as the intelligent machine develops there will be the first generation, the second generation, the third generation, and the fourth generation. That will become an important component of our national intelligence. Leaders will want to depend on this kind of intelligent machine system to act as their tool and their advisor. Of course, this kind of machine cannot substitute for people, nor can it take over the policy-making function of leaders. The machine can only be an advisor, that is clear, but without this good advisor, even more enlightened leaders will not be able to work.

A recent issue of STUDIES IN PHILOSOPHY had an article that discussed leadership decision-making [12], and the description it has about progress in modernized decision-making says that all levels of leadership should have an advising structure for decision making. Therefore, if we look at the problems from this point of view, the intelligent machine is no trivial matter, but is a matter of national concern.

1. The intelligent machine and the supercomputer are hyper- science and technology

The questions I discussed before, be they the second generation supercomputer or the first generation intelligent machine, would appear to both be matters of national concern. A scientific and technological task of this magnitude is certainly hyper- science and technology, actually, two items of hyper- science and technology. If this country is to organize and tackle key problems it must not only form a basis for the relevant science and technology, and must prepare to produce machines, but also, once preparations have been successful there are bound to be requirements and production in small amounts. Currently, as we are carrying out the requirements of our policy of opening to the world we should fully develop international exchange and cooperative activities. For example, should we participate in some of the foreign work on making the fifth generation computer? They are now eliciting international cooperation, and they would welcome our participation, because Chinese are smart and capable! We ourselves can assimilate all useable things from international exchange and cooperation. But we must soberly realize: although this is hyper- science and technology, this is like the atomic bomb, the hydrogen bomb, and intercontinental missiles, we must be independent and work hard to establish the necessary base.

2. Project Planning Questions

Of course, we must work hard and cooperatively internally as well. Our previous efforts at hyper- science and technology were led personally by

Comrade Zhou Enlai, and the experience that was successful was vigorous cooperation. Because this is an affair of national scale we need to mobilize mathematicians, psychologists, physicists, electronics researchers, optics researchers, and computer scientists, who together with computer technology specialists and researchers into thought will form a broad collective, a close-knit collective that can truly do what it says, be it second generation supercomputers or the first generation intelligent machine, both could be produced. But since this is hyper- science and technology we should operate according to the successful experience of carrying on hyper- science and technology in China over the last 30 years: unified project planning, and division into the three steps of prior research, model preparation, and finalization of design and production in arranging the overall work. For our work in particular we want to coordinate hardware and software work, for which we cannot divide people up. Supercomputers, that is, the more mature second generation of supercomputer, could be demonstrated and then a short time afterward could begin production immediately, but at the same time a concentrated effort must be made to settle the science and technology of concurrent processing. We need also to arrange the mathematics or computer math questions spoken of earlier. As for the first generation intelligent machine, based on the conditions already discussed, that is not yet ready, and we can only do preparatory research. But because it is so important we need to earnestly plan for the problems. I think that this would all require arrangement of a special conference and discussion.

3. Philosophical Questions

There is, finally, another question, which we can pose this way. Will we raise the concern of philosophers, who will say, what is this about an intelligent machine, what is this about taking the place of human labor, are we getting into mechanical materialism, into idealism?

I believe that this concern is not necessary. Machines cannot completely replace people. The so-called intelligent machine will at most do what people tell it to do. It can only replace a portion of the work of the human mind, and can only be a good "advisor." Final decisions lie with people.

Epistemologically, it is even more so. We have had the first, second, third, and fourth generation machines, and now we have the fifth. In the future there will be a sixth, and so on. Computers can only use scientific rules. Without scientific rules even people cannot function. But the scientific rules that people are aware of are only a portion of the natural world, and there is still a large portion yet unknown. This large unknown portion will need constant actual experience by people to be known, and it will require expert systems before it can be assimilated intellectually. After a few years these empirical things will rise to scientific theories, which can then again enter the realm of the computer. However, the experiences that people have not yet had cannot be assimilated by machines. Therefore, people cannot be replaced by machines, people are still people, and people are still the masters of machines.

I believe that the philosophical questions brought up here are important ones and are worth clarifying. When we used to engage in engineering technology we probably seldom involved ourselves with spiritual questions, as the things with which we concerned ourselves were material. However, the second and third generation supercomputers and the first and second generation intelligent machines, etc., that I have just spoken of all are related to human thinking, they are all concerned with spiritual and material questions. We who take up this work must also spend some effort on philosophical questions. We must still have a basic understanding of the basic principles of Marxist philosophy, dialectical materialism, and historical materialism. In this way we will avoid making mistakes in our work.

REFERENCES

1. Guo Yonghuai [6753 3057 2037], "Collected Works of Guo Yonghuai," Science Press, 1982, 22.

2. Zhang Xianglin [1728 4161 7792], Yang Chuquan [2799 2806 3123], BINGGONG XUEBAO [ACTA ARMAMENTARII], 1 (1982): 33.

3. Zhang Xianglin, "Welcoming the New Technological Revolution," National Defense Industry Publishing House, 1984.

4. Su Yucheng [5685 3558 1004], "Boundary Layer Corrections Methods in Singular Perturbation," Shanghai Science and Technology Publishing House, 1983.

5. Zhou Yulin [0719 3022 7792], Fu Hongyuan [4569 7703 3293], SHUXUE WULI XUEBAO [ACTA MATHEMATICA SCIENTIA] 1, 2 (1981): 42.

6. Gu Chaohao [6253 6389 6275], ENCYCLOPEDIC KNOWLEDGE 6 (1984): 42.

7. Caulfield, H. J., et al, LASER FOCUS 11 (1983): 100.

8. Blogett, A. J., Jr., SCIENTIFIC AMERICAN 249, 1 (1983): 86.

9. Wallace, D., NEW SCIENTIST 100 (1983): 668.

10. Ma Xiwen [7456 1585 2429], ZIRAN KEXUE [NATURE JOURNAL], 7 (1984): 409.

11. Wang Yuanyuan [3769 0337 0337], ZIRAN KEXUE [NATURE JOURNAL], 7 (1984): 446.

12. Xuan Xi [3763 2649], ZHEXUE YANJIU [PHILOSOPHICAL RESEARCH], 7 (1984): 1.

12586/9604
CSO: 4008/278

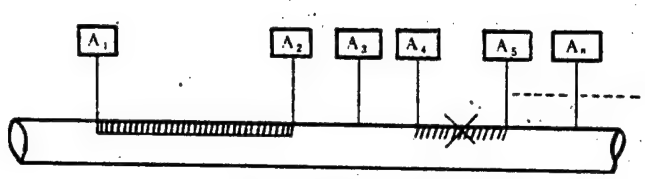
WORKSTATIONS USED WITH MULTI-PROCESSING IN DJS 26

Beijing JISUANJI YANGJIU YU FAZHAN [COMPUTER RESEARCH AND DEVELOPMENT] in Chinese Vol 21, No 4, 1984 pp 35-44, 26

[Article by Yan Youliang [0917 0645 6156] of Huabei Computer Institute: "Study of Workstation and Its Implementation With Multi-Processing in the DJS 26"]

[Excerpts] I. Problems Associated With the Bus in the Single Processor Single Bus Structure

Let us assume that there are n pieces of equipment, A_i ($i=1\sim n$), connected to a single bus of a computer (e.g., the bus of a PDP11), as shown in Figure 1. They include CPU's, storage devices and various peripheral equipment. However, at any instant the bus only allows a device A_i to actively access another on A_j ($j=1\sim n$ and in general $n\neq j$). Other equipment must wait in line to enter the bus. For example, when A_1 is accessing A_2, A_4 cannot access A_5 at the same time. In many cases, the single bus will significantly lower the efficiency of the equipment.



Single bus

Figure 1

There are several ways to improve the efficiency:

1. Increase the bus frequency.

2. Improve the bus operating mode.

3. Adopt a "connected workstation" mode to improve the utilization of the bus.

As shown in Figure 2, after adding the "connected workstation" nodes to the bus, $A_1 \longleftrightarrow A_2$, $A_5 \longleftrightarrow A_4$, $A_5 \longleftrightarrow A_i$. We will mention later that multi-processing in the DJS 26 is achieved by two highly random time sharing buses.

However, as a system sharing all the resources, the CPU and storage devices must be connected with all peripheral equipment. But, because information transmission rate is lower in the peripheral equipment than in the CPU, the requirements are quite different. There are different kinds of devices with various response times which are at different distances away from the CPU. Because of the long distance, number of devices, and powerful electrical equipment involved, the signal reliability requirement is even higher. Therefore, the main concerns in designing this type of bus are capacity, transmission range, reliability, and independence of equipment distance and response time.

These requirements obviously are not compatible with high speed and high efficiency. This is a serious problem in a multi-processing system with a bus structure. It is necessary to at least design two different buses to meet these requirements. In addition, these two buses must be connected ($i=6 \sim n$) for simultaneous accesses. However, when $A_1 \longleftrightarrow A_4$ happens, A_2 and A_3 cannot simultaneously access any equipment because to access any equipment they must occupy a common overlapping bus. However, it is possible to simultaneously access $A_i \longleftrightarrow A_j$ (i, $j = 5 \sim n$ and in general $i \neq j$) because they do not share the same overlapping bus. The "workstation" may have some isolation effect.

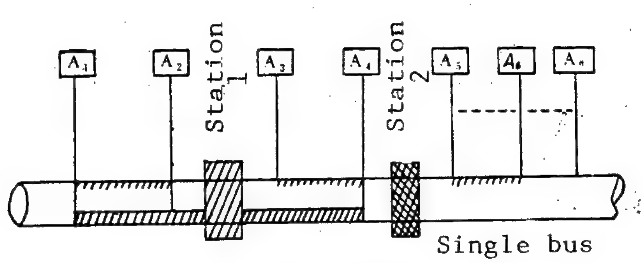


Figure 2

II. Problems Associated With the Bus in a Multi-Processing System

In a single bus or multi-bus system, information exchanged between a central processor and a storage device must be transmitted through the common bus because all processors are operating at high speeds. Therefore, the bus must have a high transmission rate. The primary requirements for this type of bus are high speed and high efficiency. Furthermore, its logic function must be consistently maintained (not the operating mode or electrical property). This can be realized by connected "workstations."

Consistency in bus logic is equivalent to that a "workstation" is logically transparent. The presence of a "workstation" will not be felt from the viewpoint of its logic function. Either bus can be considered as an extension of the other.

Because a "workstation" has the above characteristics, it can be used to create large-scale multi-processor systems. In this case, a "workstation" can be considered as a node. It is possible to design a "workstation" with n "iron arms" (see Figure 3 a, b). As long as there is no conflict, any path may be connected to another. When there is a conflict, the signal train may be temporarily stored in a storage device on the side. This information will be re-transmitted when permitted.

Figure 3 shows the schematic diagram of a node with n=3.

A "workstation" with an n=8 node can be used to create a high performance system, as shown in Figure 4. Each node S_{ij} ($i=1\sim n$, $j=1\sim m$) is connected to a CPU_{ij}, internal storage INM_{ij}, external storage EXM_{ij} and peripheral equipment EXD_{ij}. Thus, a superstructure with nxm CPU, INM, EXM, and EXD can be formed. (Note: The number of each type of equipment is not limited to one. It is not difficult for the reader to deduce the number of connected equipment in this generalized case.) Each CPU_{ij} has the capability to access INM_{ij}, EXM_{ij}, and EXD_{ij} at a high speed without affecting the function of the equipment at other nodes. It acts as an independent system. In the meantime, it shares all the resources of the entire system. This is realized by connecting the nodes.

For a piece of equipment at S_{ij} to access another at $S_{i+k,\,j+1}$ it is necessary to go through k+1 "workstations" in k+1 steps. Without any conflict, the time delay is t(k+1) where t is the time delay at each "workstation."

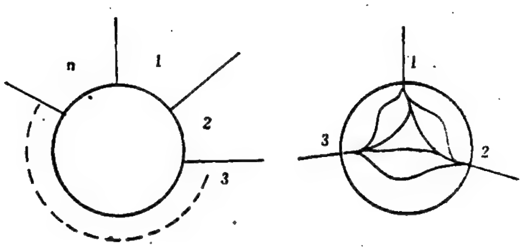


Figure 3. (a) A node with n lines
(b) Structure inside a three line node

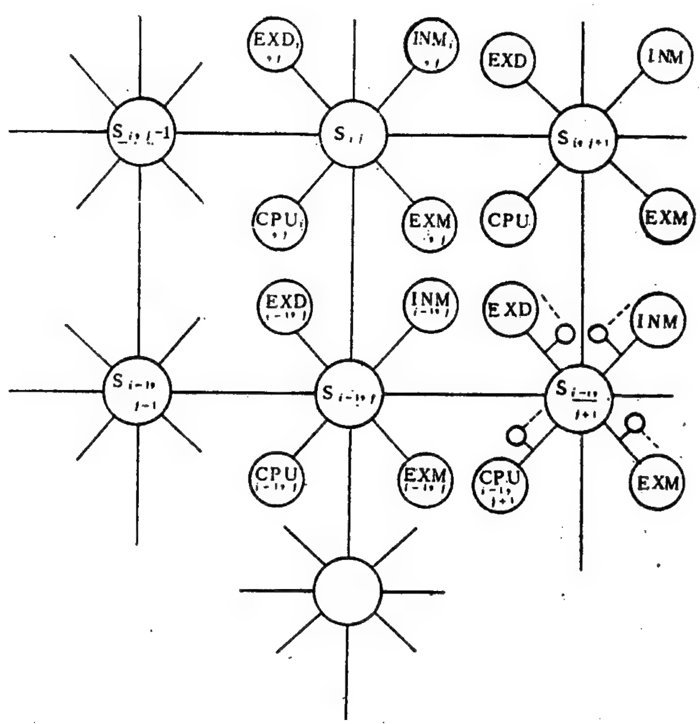


Figure 4

In medium and small integrated circuits, there will be too many devices when n is large. For example, if n=8, then S=56. Therefore, we must also consider the present technology and the superiority of the two "random prediction timeless buses" in the design of the DJS 26 multi-processing system. We used four "workstations" and connected each to one external bus and two internal buses. There are two parking lanes in each "workstation" as specified in the design described later.

If we consider the DJS 26 multi-processing system as an element, its external connection is shown in Figure 5 (a). The four "workstations" have four arms to link with other elements, in analogy to the four carbon bonds in organic chemicals. In particular, they may be connected to themselves to form a supersystem.

Thus, in principle, DJS 26 may form a supersystem such as the one shown in Figure 5 (b). Note that there are two one-way lines, similar to a two-way line, but not two channels. This is to prevent the system from locking itself.

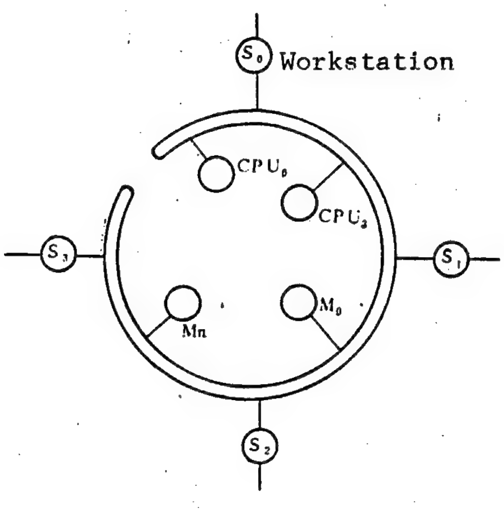


Figure 5 (a)

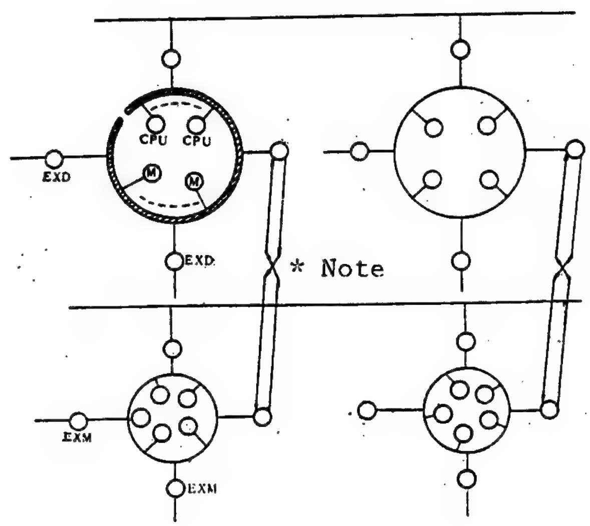


Figure 5 (b)

In reality, this type of connection will require the expansion of the "workstation" address and some local technical modification. The realization of the "workstation" in the DJS 26 multi-processing system will be discussed in the following:

III. Function and Significance of Workstation in DJS 26 Multi-Processing System

In a DJS 26 multi-processing system, as shown in Figure 6, one end of a "workstation" is connected to two highly time-sharing random buses, i.e., internal buses. The other end is connected to a high capacity timeless responding bus, i.e., external bus.

The following functions are accomplished in a "workstation":

(a) matching different operating rates in two buses;

(b) switching different operating modes in two buses;

(c) expanding load capacity and transmission capability;

(d) enabling all buses to operate simultaneously when there is no overlap;

(e) switching all interrupting signals to peripheral equipment to reduce waiting time for information on queue.

These functions are obviously valuable. Devices of different speed requirements, operating modes, and distances are connected through coordination.

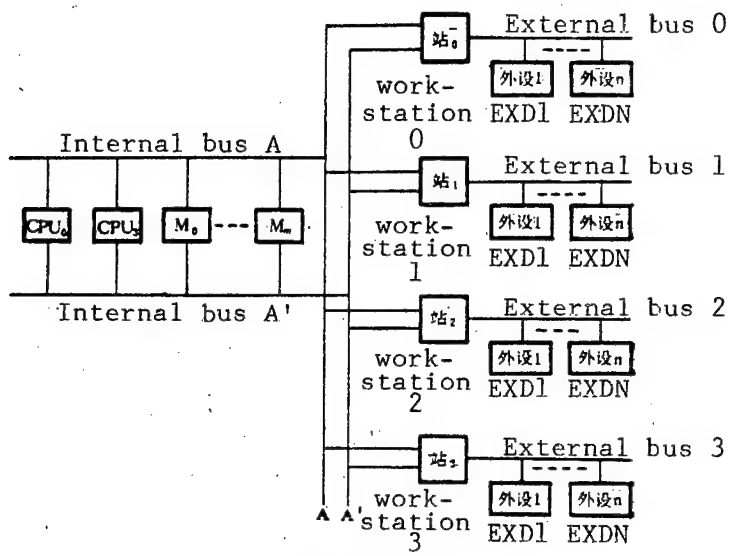


Figure 6

A "workstation" is an accessable piece of equipment with respect to the internal bus. It may be considered as a storage device. As for a device capable of asccessing other devices on its own, it can be considered the same as a CPU.

A "workstation" is the first piece of equipment on the external bus. It not only can actively access other devices but also can be accessed by other devices.

IV. Operating Principle of a Workstation in a DJS 26 Multi-Processing System

As shown in Figure 7, the structure and operating principle of a "workstation" is similar to that of a railroad station. In the design, a "workstation" is only connected to three lines and two of them are used for parking only.

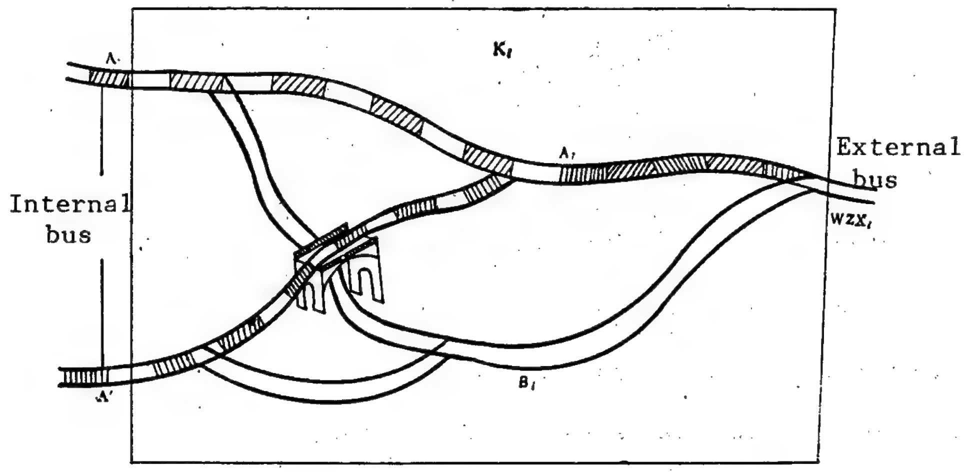


Figure 7. Operating Principle of a Workstation

In structure, a "workstation" consists of two parking lanes A_i and B_i and a dispatcher K_j. The parking lanes are connected to the high speed line (internal bus) and slower line (external bus). The dispatcher K_i is responsible for assigning the time to enter the station, the lane to park and the time to leave.

The four operating states of a "workstation" are described below:

1. CPU_j requests to transmit data to the nth device on WZX_i through "workstation" Z_i. This is similar to the process of shipping a fully loaded railroad car from CPU_j on the high speed railroad A through the station Z_i and finally reaching the outfit n via the WZX_i branch (see Figure 8).

2. CPU_j gets data back from the nth equipment on WZX_i through station Z_i (see Figure 9).

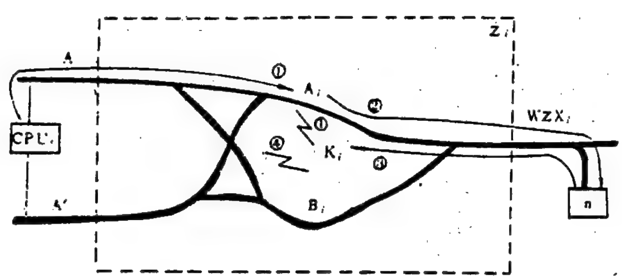


Figure 8

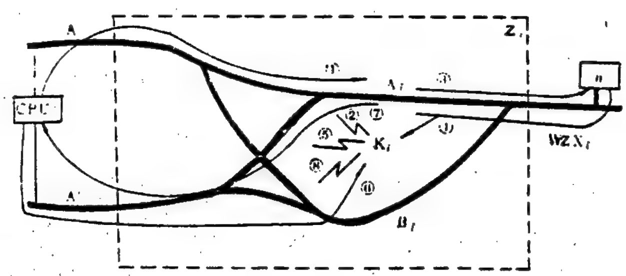


Figure 9

3. Data from an equipment (such as memory M_j) on the internal bus is entered into peripheral equipment n (see Figure 10).

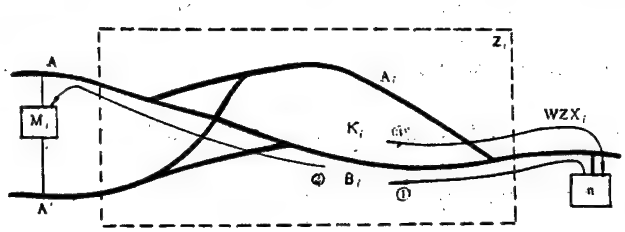


Figure 10

4. The process for data in an equipment on the external bus (e.g., storage device M_j) to be read by a peripheral equipment n via station Z_i is as follows:

(1) The peripheral equipment n first requests to use the external bus WZX_i. After the request is approved by the dispatcher A, an empty train enters the WZX_i track. Before entering Z_i, K_i checks the pass. If the pass is valid, the empty train may enter Z_i and park along track B_i.

(2) After Z_i receives permission to transmit signal from M_j, it requests to use A. After approved by the dispatcher at A, the empty train is moved from B_i to A and then goes directly toward M_j. The transport on A is thus completed. The empty train is loaded at M_j upon arrival.

(3) As the transport on A is terminated, under the control of K_i, the state of Z_i is changed to waiting for goods to arrive.

(4) In the meantime, M_j will cancel its permission to transmit signal so that other equipment on A cannot send any more information into M_j.

(5) After the loading (output) is complete, M_j will send a signal to every CPU and Z to indicate that the loading is done and it is ready to be picked up.

(6) Because only Z_i is waiting for goods to arrive from M_j, therefore, only Z_i will immediately request to use the high speed railroad A' after receiving the signal from M_j. When the situation permits, a signal is sent to M_j through the dispatcher at A'.

(7) After receiving the signal, M_j will move the loaded train to track A'. The train goes directly to Z_i and is parked on the track B_i. In addition, the occupation of A is terminated and M_j is reset to permit delivery.

(8) Under the control of K_i, the train is moved from track B_i to the slow track WZX_i to go directly to n (Note: B_i is already vacated). Finally, a termination signal is sent.

(9) After n receives the goods and the termination signal, it gives up the right to use WZX_i.

The process to read data is described in full and the diagram to explain this process is omitted.

V. Logic Principle

The operating principle of a "workstation" has been described in detail earlier. The logic principle is briefly discussed here. There are the following signal lines at the junction between the internal bus and Z_i (relation between a "workstation" and the bus, see Figure 11):

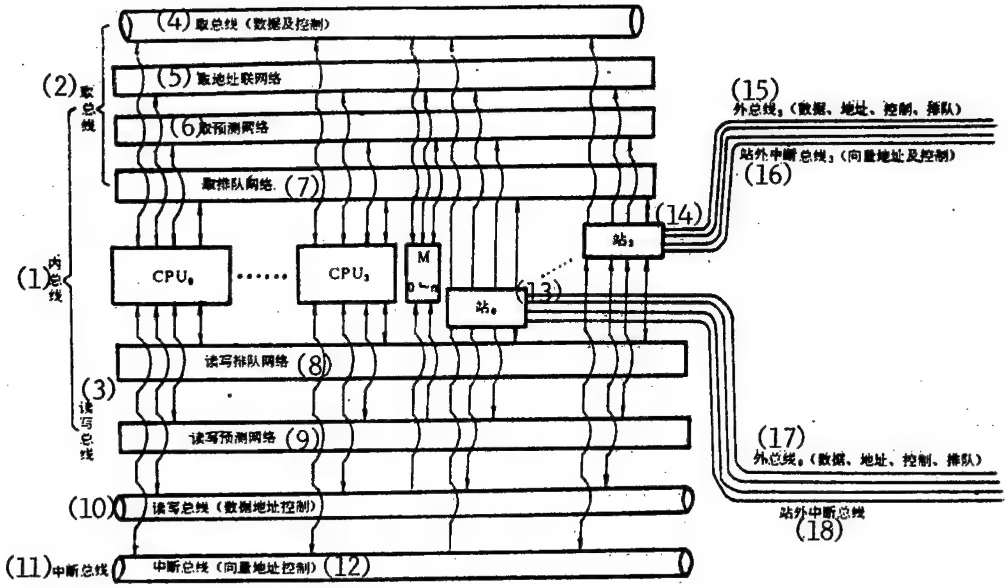


Figure 11. Relation Between Internal Bus and Workstations

Key:

1. Internal bus
2. Fetch bus
3. Read/write bus
4. Fetch bus (data and control)
5. Fetch address network
6. Fetch prediction network
7. Fetch queue network
8. Read/write queue network
9. Read/write prediction network
10. Read/write bus (data address control)
11. Interruption bus
12. Interruption bus (vector address control)
13. Station_0
14. Station_3
15. External bus (data, address, control, queue)
16. Interruption bus outside workstation (vector address and control)
17. External bus (data, address, control, queue)
18. Interruption bus outside workstation

Two examples (A and B) are given below to explain the logic principle of a "workstation" (see Figure 12).

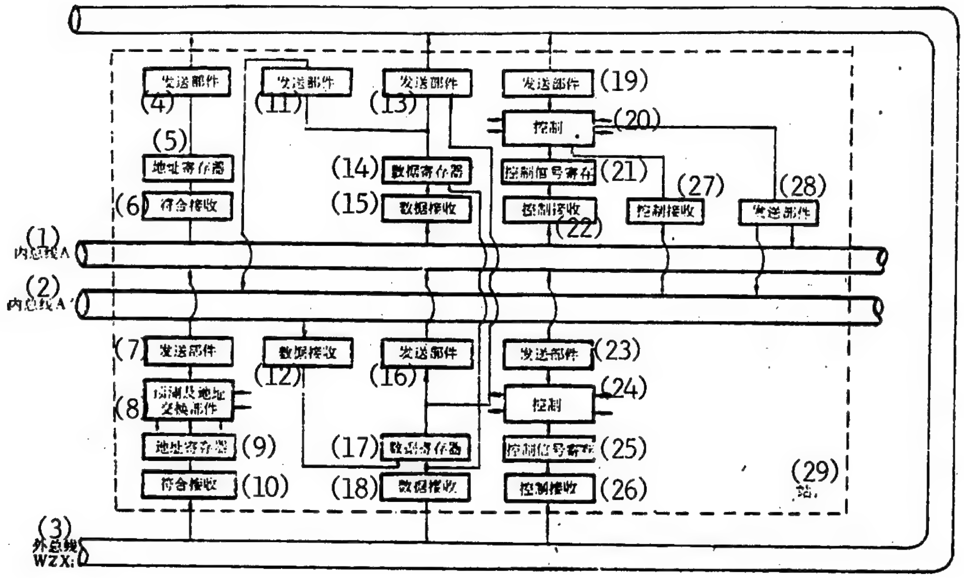


Figure 12

Key:

1. Internal bus A,
2. Internal bus A'
3. External bus WZX_i
4. Transmitting parts
5. Address register
6. Check and receive
7. Transmitting parts
8. Prediction and address exchange elements
9. Address register
10. Check and receive
11. Transmitting parts
12. Receiving data
13. Transmitting parts
14. Data register
15. Receiving data
16. Transmitting parts
17. Data storage
18. Receiving data
19. Transmitting parts
20. Control
21. Control signal register
22. Receiving control
23. Transmitting parts
24. Control
25. Control signal register
26. Receiving control
27. Receiving control
28. Transmitting parts
29. Station i

A. The process for CPU_j on the external bus to access a peripheral equipment n to read data is as follows:

(1) CPU_j sends a request to the internal bus A after an "I/O line" is established at a workstation. Upon approval, CPU_j sends the address and control signals to A (300 ns).

(2) The "workstation" address is checked against the line address to determine whether it is under the control of the station.

Note: The operating signal arrives 75 ns late and ends 25 40 ns early to avoid error due to signal distortion.

(3) If everything checks out, then the address and control signals are stored in address and control registers in the "workstation." In addition, the trigger for accessing WZX_i is set to "1." The internal bus is automatically terminated so that it may serve other equipment on a time-sharing basis. CPU_j is in the input mode. The "I/O prediction" of Z_i is cancelled so that it cannot be accessed for I/O.

(4) If WZX_i is not used by other equipment, the freeze line must be open (at high voltage). If WZX_i is occupied, then we have to wait until the access is over. The request will be entered into a queue trigger.

(5) When the queue trigger is "1," a freeze signal is generated to freeze all the requests to enter WZX_i. No new requests after the freeze are allowed to enter the queue trigger (because many devices may request to access WZX_i).

In the meantime, the queue signal is sent from the "workstation" according to the WZX_i junction sequence, from a near device to the next one down the line. Experimentally, it was proven that the transmission range could reach 50 M (joining 8 devices). It is estimated that joining 20 devices is feasible. The time delay in this case is 500 ns. Therefore, there is a 500 ns stabilizing circuit to ensure the timing stability of the queue signals.

(6) After the queue transmission is stabilized, WZX_i must be occupied by a "workstation" because of priority. The "workstation" transmits information in the address and control registers to WZX_i and opens the relevant gates in the data register under the control of the I/O control register.

(7) When a peripheral device receives signals from WZX_i, the address is checked. Furthermore, the corresponding data is transmitted to the WZX_i data line under the control of a read signal (if n is a storage device, it will be read first). n then stops operating and sends a responding signal to WZX_i.

(8) After Z_i receives the responding signal, the leading edge is used to send the data into the data storage device in Z_i. Note: the responding signal and other signals are maintained so that WZX_i remains in an ccupied state. This responding signal resets the output trigger on Z_i.

(9) Under the control of the read-out trigger, Z_i sends a pre-fetch signal.

(10) After CPU_j receives the pre-fetch signal from Z_i, the access address coincides with the pre-fetch signal because it coincides with the address in CPU_j and also because only CPU_j is in a fetch state with respect to Z_i. In this instance, CPU_j will request for A'. Upon approval, CPU_j will send a fetch signal to Z_i.

(11) This fetch signal will open the pertinent I/O gates to transmit the data in register A_i to the data line on A'.

(12) CPU_j will send the data on A' to a storage device S.

(13) In the meantime, the leading edge of the signal is used to reset the read-out trigger and the queue trigger in order to remove the pre-fetch signal. When the fetch signal is in existence, the request and queue signals are maintained until the trailing edge of the fetch signal ends.

(14) After the trailing edge ends, address and control signals are eliminated and WZX_i is released. Until the responding signal vanishes, the frozen line is released in order to re-establish the I/O prediction signal of Z_i.

(15) After the disappearance of the WZX_i address and control signals have been transmitted to n, the responding signal from n can then be eliminated because response signals are supported by the address and control signals.

(16) After the cancellation of the responding signal reaches Z_i, the "read-write prediction" signal is re-established.

B. The process for an equipment n on the external bus WZX_i to read data from a device M_j on the internal bus through a "workstation" is as follows:

(1) Equipment n requests for use of the external bus (to set the request trigger to "1"). It enters the queue trigger when the freeze line of WZX_i is open.

(2) After the queue trigger is set at "1," a freeze signal is generated to lock in the freeze line. A queue signal is also generated to create a serial queue. Moreover, a 500 ns stabilizing circuit is set at "1" to ensure the stability of the queue transmission.

(3) After the queue transmission is stabilized, address and control signals are sent to WZX_i.

(4) After coinciding the address on Z_i with the address signal on WZX_i, the address and control signals received are stored in the address and control register B_i in Z_i. Furthermore, the trigger for Z_i to access the internal bus is set to "1."

Note: n will remain occupying WZX_i until the responding signal is over.

(5) After Z_i receives the "read/write prediction" signal from M_j, it enters a queue for the internal bus A under the condition that the freeze signal of A is open. This process is repeated if the queue is not successful. If successful, it will occupy A.

(6) When Z_i occupies the internal bus, the relevant address and control signals are sent to the internal bus A. If the system assumes that the "workstation" operates in mode N to access the storage device, then the mode is controlled according to the mode register and mode control switches.

(7) After checking the address and control signals arriving at M_j, they are entered into relevant registers. In the meantime, Z_i is switched into a fetch

state. At this instance, the "read/write prediction" signal will disappear automatically. After a 300 ns cycle, Z_i will automatically give up A. Z_i also opens the data receiving gate for B_i.

(16) After Z_i receives the cancel message from address and control signal, Z_i cancels the "respond" signals.

(17) At this point, Z_i returns to its initial state. WZX_i also completes the transmission cycle. For now, the query is over.

VI. Time-Out Problem

When a CPU accesses a peripheral equipment through a "workstation" for a long time, then the bus will be congested and the CPU cannot function. For this reason, we have to take proper measures to overcome this problem. The internal bus itself will not be congested because a single stabilizing cycle is fixed at 300 ns. A time-out control circuit is used in the CPU. When an equipment cannot be accessed for a long time, the CPU will automatically enter a capture state for the system to process this matter.

Because it is necesary to go through a workstation and two buses for a CPU to access an equipment on the external bus, therefore, the CPU will consider that the access is complete once the access to the internal bus is successful (i.e., access to the "workstation" is successful). In reality, a "workstation" must still continue to access the peripheral equipment (e.g., in the case of write). Sometimes the access may not be successful and thus creates a "time-out" situation on WZX_i. This time-out will be registered by the time-out trigger in the "workstation" to create an interruption. When the interruption is not responded, this path will be closed.

Hence, as long as the "workstation" time-out value is longer than that of the CPU, then the time-out caused by the access (including absence of equipment) will be discovered by the CPU as $T_Z^A > T_{CPU}$.

However, even a time-out occurs when the CPU is writing, this time-out will be spotted by the system. However, because the address accessed is not kept in the "workstation," it is not possible to use a simple method to determine the equipment. It may be determined by special software.

When a time-out occurs during the access of a peripheral equipment to a device on the internal bus through a "workstation," the time-out should be registered on the equipment n. Hence, as long as the time-out value of Z to access the internal bus is set to be less than that of equipment n, $T_Z^B > T_n$, this function can be realized.

12,553/9599
CSO: 4008/1019

ZB-792 SOFTWARE SYSTEM FOR GRAPHIC EDITING OF LSI PATTERNS

Beijing JISUANJI YANJIU YU FAZHAN [COMPUTER RESEARCH AND DEVELOPMENT] in Chinese Vol 21, No 4, 1984 pp 45-51

[Article by Hong Xianlong [3163 0341 7893] of Qinghua University, Zhong Longbao [6988 7893 0202], Xu Qinglin [1776 1987 2651], and Zhang Wentao [1728 2429 3447] of the Automation Institute of the First Machine Building Ministry]

[Excerpt] ...The operating system RDOS is supported by either the DJS-130 or the NOVA machine.

I. Introduction

As technology and materials are developed, it is currently possible to manufacture very large scale integrated (VLSI) circuits with tens of thousands of elements on a single chip. One of our tasks is to see that the CAD technology promoting the growth of LSI also meets the needs in VLSI.

Computer-aided design and manufacturing of LSI and VLSI masks is an important component of the CAD technology. Because of the limitation of automated design software and the high yield requirement for chips, most of the work is still done manually. Then, the manually designed pattern is entered into a computer through a man-machine interactive or descriptive graphic software to obtain the master mask of the chip after editing.

The ZB-792 system is a language descriptive graphic editing software system for LSI and VLSI, developed based on ZB-781 and ZB-791 systems. This system operates based on the support of RDOS for the needs of data processing in VLSI patterns. The ZB-792 system was certified by Qinghua University at the request of the Fourth Machine Building Ministry in October 1981. This system was used to make more than a dozen integrated circuit masks, including the 4K static NMOS RAM, E/D RAM, 1K NMOS RAM, etc. Our experience shows that the system is not only working but also works efficiently.

ZB-792 inherited certain advantages of ZB-781 and ZB-791. For example, the software is modular and a multi-path editing technique is used. It adopts a data network structure to reduce redundancy. Certain optimization methods are employed in the output program to minimize the operating time. In addition, it fully utilizes the resources of RDOS and uses disk files and the virtual

buffer technique to render the mask data unlimited. Disk files are used as a medium to connect this system with the man-machine interactive graphic editing system, the design-checking software, the circuit layout software, and the mask graphics data base. Through continuous expansion, it serves as a base for a perfect LSI CAD system.

In this article, we will first discuss the requirements of a graphic editing software for VLSI. Then, we will introduce several features of ZB-792 in meeting the editing needs for VLSI masks.

II. Special Features of VLSI Pattern and Requirements of Graphic Editing Software

There are tens and hundreds of thousands of elements on a VLSI chip. Therefore, the mask of the chip has numerous patterns which are densly packed. In order to effectively utilize the area on the chip and to minimize the chip area, in addition to reducing the line width, spacing, and pattern size to the extent allowable by technology, many non-orthogonal patterns and slanted lines are used to make the mask more compact. Figure 1 shows the leakage grid of a store element of a 4K MOS RAM, which requires the graphic editing software to be able to process non-orthogonal patterns as well as circles and sectors. However, orthogonal patterns (including rectangles, polygons, and hollow polygons) are the majority by proportion. This feature must be considered in software input language, data structure, and pattern calculation in order to reduce the work load and to improve the efficiency of the computer.

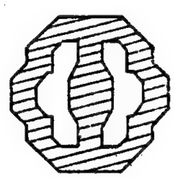


Figure 1. An Example of a Non-Orthogonal Pattern

Next, because the amount of graphic data is enormous, we must also consider how the data structure and file structure of the graphic editing software can meet the needs of VLSI in order to obtain satisfactory results in conserving storage space and ease of retrieval.

Third, a VLSI mask may contain up to millions and tens of millions of patterns. The number of coordinates is even larger. It is impossible to read these coordinates manually. One of the primary tasks for the designer of the graphic editing software is to minimize the amount of manually prepared data to the extent acceptable.

Fourth, because of the vast amount of data of various kinds, human error in data preparation is almost unavoidable. Therefore, to check the data for accuracy is also an important task.

Finally, the LSI CAD process is a comprehensive one. How the graphic editing software is linked to other CAD softwares and data bases is also of great concern.

III. Special Features of ZB-792 Graphic Descriptive Language

There are three types of ZB-792 graphic descriptive language: graphic element, geometric transformation and combination, and definition and utilization of builder blocks. It is compatible with ZB-781 and ZB-791 languages. The necessary expansion was made based on the latter to meet the needs of VLSI.

By taking into account that the majority of the patterns in an integrated circuit mask are orthogonal, there are many phrases in the language describing various orthogonal patterns, such as two phrases for rectangles (using two diagonals and using the lower left corner and the length and width), polygons, hollow polygons, equi-lateral polygons, overlapping rectangles, and upper platform pattern (designed for special users). For non-orthogonal patterns, the language provides phrases to describe a polygon forming "broken line," an oblique parallelogram and a line connecting multiple layers (allowable at 45°). If required, there are provisions to include circles and sectors. These graphic elements can meet the needs in making the masks for LSI and VLSI. In order to minimize the work load of the user in data preparation, these phrases keep the amount of information to describe these patterns to a minimum. For instance, it is believed that in most cases the two neighboring sides are still perpendicular in a polygon, the language takes every other coordinate on both sides of the right angle, instead of every point. The language for the polygon shown in Figure 2 is:

$$\#ZX\ x_1,y_1+x_2,y_2+x_3,y_3,x_4,y_4+x_5,y_5+x_6,y_6,x_7,y_7-x_8,y_8,x_1,y_1 \swarrow$$

in which #ZX is the symbol for a broken line, "+" represents taking every other point on both sides which are initially perpendicular then parallel, "-" also represents taking every other point on both sides, however, they are initially parallel and then become perpendicular, and "$\swarrow$" is the end of the language. In addition to using "+" and "-" to relate a coordinate pair, two coordinates connected by "," become a point by point pair. Thus, a great deal of information is saved. Figure 3 shows a metal wire which can be written in either a broken line phrase or a polygon phrase. However, because it is equi-lateral, it is also possible to write in it a corner phrase. The first two phrases will require 9 or 10 coordinate pairs, while the latter only needs 5. The required coordinates are shown as x_i and y_i in Figure 3.

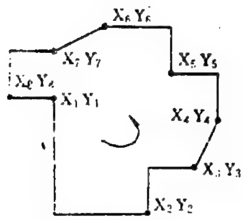


Figure 2. An Arbitrary Polygon

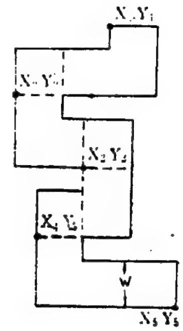


Figure 3. An Arbitrary Shape With Sharp Corners

Geometric transformation and combination languages include eight types of transformation such as translation, symmetry and rotation, and pattern repetition and hashing. It is very convenient to use these phrases to describe an integrated circuit. Furthermore, it can drastically reduce data preparation. In a VLSI or LSI mask, a more effective technique is the definition of building blocks and modular capability. This is designed for the "from bottom to top" building block method.

Usually, an LSI chip is comprised of several functional blocks. Each functional block consists of more smaller blocks....down to elements such as gates and triggers. Quite a few of these elements and functional blocks are identical. Some of them may be standard or common elements with existing patterns. Some may have to be designed. The smaller ones are designed first. Lower level blocks are assembled together in designing a higher level block. These elements and blocks are defined as building blocks and their source programs are written in ZB-792 language. Defined building blocks can be used in preparing a higher level block and a chip pattern. If necessary, geometric transformation, pattern repetition, and hashing can be made. Moreover, these blocks can be used as modules repeatedly. This language structure not only significantly reduces the information to form the mask but also has advantages such as structuring, ease of modification, and ease of checking. It also is a vehicle for elemental graphic input in automatic design.

In addition, ZB-792 has built-in array language for the automatic design of PLA, ROM, and decoder circuits. The patterns of these circuits are used to automatically generate masks based on limited information provided by the user.

Our experience shows that the abundant language description capability, modular structure, and array language for automatic design of ZB-792 can meet the needs in LSI and VLSI.

IV. Network Data Structure

The amount of data contained in a VLSI pattern is very huge. It often includes millions of diagrams. Direct storage of each diagram not only takes up more space but also consumes more computer time. Furthermore, it is

inconvenient to perform design and logic checks. A network data structure is used to store graphic data in ZB-792. Each building block is a data file and files are linked by a pointer. The same block is only stored once.

The relation between the overall pattern and the blocks for a chip structure such as the one shown in Figure 4 can be expressed by a tree structure shown in Figure 5. Each building block and the overall pattern are stored as files and these files are in a network structure as shown in Figure 6 (the last KA_3 on the the right should be KB_3). A more general network structure is shown in Figure 7.

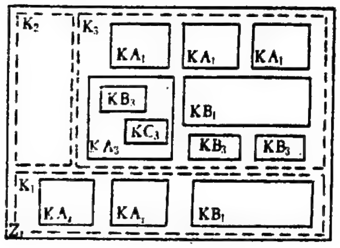


Figure 4. Chip Structure

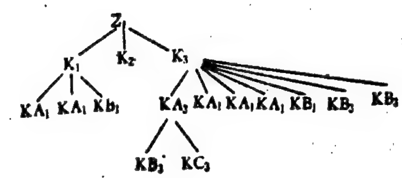


Figure 5. Tree Structure of the Chip

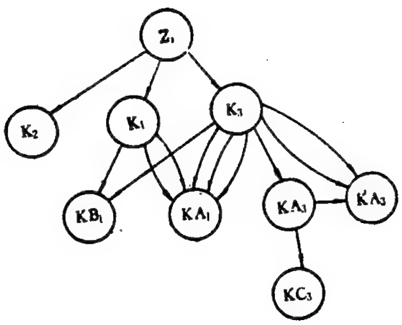


Figure 6. Network Data Structure for Chip Shown in Figure 4

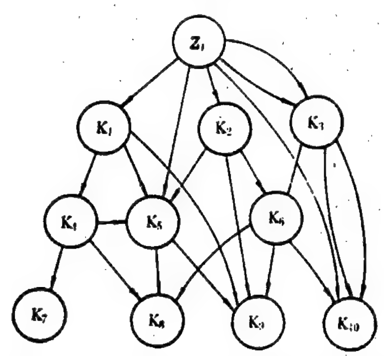


Figure 7. A General Network Structure

From Figures 6 and 7 one can see that each block only has to be stored once. There is no redundancy. This type of structure significantly reduces the demand for storage space, which is generally one-hundredth or even one-thousandth of the amount of data contained in a graph.

Graphic data is usually stored as a disk file, considering the need for frequent modification. Hence, each block file is stored in a serial document structure. Editing the pattern in a block will not affect the address of other data to save processing time. The entire graphic data file is stored according to an index file which gives the entrance of each block file for quick access.

V. Pattern Display and Plotting

To display and plot a pattern is an imperative function of graphic editing software. It is often used as a technique for checking and modification. It is very difficult to avoid mistakes when thousands of patterns are prepared manually. In addition, mistakes may be made due to design error. Therefore, graphic checking becomes even more important.

There are two ways to check a graph. One is to use specially developed software to determine whether any design error exists. (We have already developed a graphic checking software, JC-81, which will be introduced in a separate article.) The other one relies on visual inspection of the display and plot. Due to various reasons, software may not be able to find all the errors. In this case, the display or plot is needed. In fact, a user always checks the pattern manually to ensure its accuracy before a mask is made. In general, display is used in partial revision and a plot is used in overall checking and document filing.

ZB-792 offers 15 display commands for checking and editing. These 15 commands are: open window, restore window, move window, display grids, display coordinate values, change display shape, delete pattern, change mask level number, graph with optical pen, graph with keyboard, split frame display, change display format (with and without overlapping edges), clear graph, erase screen, and display complete and return to program.

The display program is realized on a TEKTRONIX 4014 storage graphics display. The screen is divided into two parts, graphic and command areas, to allow the display of graphics to function in an alternating mode. As a user issues a command, the command as well as the graph are displayed at the same time. It beeps if a wrong command is issued.

The display program offers various function-checking commands to the user so that he can display a certain layer of the pattern on the screen. Moreover, it can also simultaneously show several layers which can be checked through the overlaps between layers. In order to distinguish each layer, the line type command can be used to specify the display line type for each layer (such as solid line, dotted line, etc.). The open window function in the program allows a user to enlarge a graph to the full screen. An enlarged graph can further be enlarged. This multiple window opening may be eight levels deep. After a user issues an open window command, the two diagonal apex coordinates are entered either by positioning a cross hair or from a keyboard to complete an open window command. Because a fixed point fast tailoring method is incorporated in the software, the command can be executed within several seconds. Afterwards, the window moving command can be used to focus on another area of interest. Or, the restore window command may be used to return to a previous screen. These window functions enable the user to easily locate the area of interest for observation.

There is a coordinate grid command to provide a reference coordinate to the user. It appears as a transparent grid paper on the screen. The "display

coordinate" command offers convenience to the user who wants to know the coordinate of any point. Thus, a user not only can see the shape and relative position of the pattern but also knows the size and spacing of each pattern. This is very important to ensure the accuracy in graph checking.

The "split frame" display capability can automatically divide the graph into equal regions according to user specification. Each region can then be enlarged to full screen one by one from bottom to top and left to right for observation. As shown in Figure 8, the entire graph is divided into 16 regions after the user issues the split frame display command and specifies the size of the region with an optical pen (as indicated by Point A in Figure 8). The advantage of this software is that the window is automatically moved to allow the user to view the entire graph. If a hard copy device (a TEXTRONIX accessory) or a camera is available, then a complete graph can be put together much faster than using a plotter.

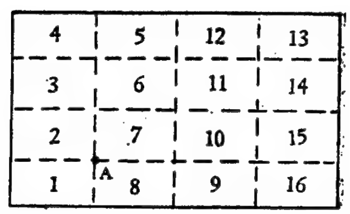


Figure 8. Schematic Diagram of Split Frame Display

As the degree of integration rises, the amount of graphic data also increases drastically. It usually takes a long time to show the entire graph on the screen. For example, the entire metal leakage grid area of a 4K static NMOS RAM takes 30-40 minutes to display. The pattern displayed is too congested to see clearly. It only provides a window position for the next level of display. The ZB-792 display program avoids this unnecessary display. Before a display command is entered, it shows the outer frame of the graph for the user to open the window. Once the command is issued, the system quickly sends the picture in the window to the main display in several seconds. The response time of the display is significantly improved. A user may not know the pattern distribution with a blank frame, which makes it difficult to choose the window position. This problem is solved by the "schematic display" format. In this mode, all the patterns in the blocks are omitted. The languages associated with the elements remain displayed. A single line is used to connect two blocks. Thus, the user can understand the overall layout for selecting a suitable window position. This takes only several seconds.

Plotting is not only a way to check the diagram but also a measure to file the document. In addition, engraving is also a technique to make masks. Usually, plotting is done off-line. The computer will record the plotting data on tape to be mounted on the plotter controller. When plotting data is processed, the entire diagram is broken down into lines and then recorded on the tape. The work load is very large. The imported XYNETICS graphic software package

includes FORTRAN subroutines for data processing and recording. It usually takes several hours to finish. For instance, it takes several hours to record the first level of a 4K NMOS RAM. The ZB-792 plotting program uses a directly compiled language. Because a fixed point method is used, the processing speed is raised dozens of times. It only takes 13 minutes 23 seconds to record all the data (7 levels) for a 4K NMOS RAM. The system also offers many features such as window opening, enlargement, rotation, and arbitrary origin to improve its flexibility.

VI. Output Modes

Usually, an integrated circuit mask is prepared by an optical graphics generator, an electron beam machine, and a plotter. Furthermore, there are many models which require different interfaces. Even the graphic editing software must accommodate their specifications, it if difficult to make it universal. ZB-792 accommodates the specifications of various equipment and offers the proper interface for a specific device.

Through a user command, ZB-792 offers two editing and two output modes. The two editing modes are to divide and not to divide regular polygons into rectangles. The two output modes are to eliminate and not to eliminate overlapping borders. When overlapping borders are eliminated, line segment queuing and out sequence optimization are also done at the same time. The polygon ABCDEFGHIJ shown in Figure 9 is cut into four rectangles AKIJ, KBMH, CLGM, and LDEF. In the uncut editing mode, the border of the polygon is stored in the computer. In the cut mode, four rectangles are stored. When four rectangles are stored in the computer, three overlapping sides IK, MB, and FL are drawn in the output mode in which the overlap is not eliminated. In the other mode, the original polygon is obtained. In addition to using a cutting mode in editing, other reasons, including methods used in initial data preparation, may cause overlaps. There are four combinations of editing and output modes:

A. cut + eliminate

B. uncut + eliminate

C. uncut + uneliminate

D. cut + uneliminate

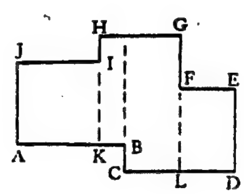


Figure 9. Example of Overlapping Edge

(There should be a point M between H and G.)

Mode A does not include any overlapping borders even when a user enters a regular polygon as several rectangles. But, it takes longer on the computer. Mode B will not result in any overlaps if the user enters all regular polygons as such. The result of Mode C is identical to that of the input mode. In addition to meeting the limitation of Mode B, we have to avoid using large

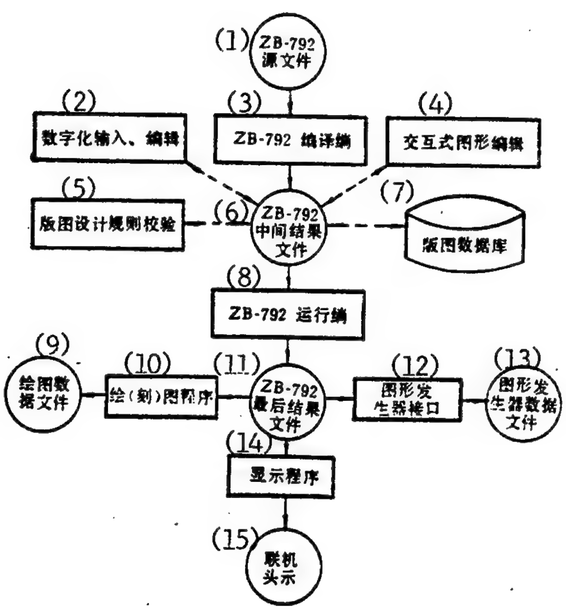
rectangles and level linkers to eliminate overlaps in the output. Mode C takes the least amount of time. These three methods are primarily for plotting the pattern which only requires the outline of the pattern. A user may choose one of the three modes according to his need.

Mode D results in rectangular data. Its processing rate is also fast. It is suitable for graphic generators and electron beam mask makers because they require the actual pattern, instead of an outline. The four rectangles in Figure 9 is equivalent to the polygon. The exposure unit of a graphic generator is a rectangle.

ZB-792 can write the output data on tapes or on disks, as chosen by the user.

VII. Connection to Other CAD Software and Data Base

As the degree of integration, the complexity of the circuit and the design data increase, a higher reliability is required. It is inadequate to rely on a single software for CAD and CAM. An auxiliary graphic design system centered around a graphic data base is needed. Considering the limitation of a small computer in memory and speed and the applications in China, we are primarily using manual design. A descriptive and interactive graphic editing software is used for input, assembly, and editing. A graphic design guideline software is used for checking. ZB-792 is a descriptive graphic editing software which works in two steps: editing and operation. The editing step converted the source program to a disk file in "intermediary codes." The operating step converts "intermediary codes" into final codes. Because the data in intermediary files are stored in a network structure, it is highly structured to save storage space. It is a "medium" to link to other CAD software and data bases. The structure of the entire system is shown in Figure 10.



Key:

1. ZB-792 source file
2. Digital input, editing
3. ZB-792 editing step
4. Interactive graphic editing
5. Graphic design check
6. ZB-792 intermediary files
7. Graphic data base
8. ZB-792 operating step
9. Graphic data file
10. Plotting program
11. ZB-792 final files
12. Graph generator interface
13. Graph generator data file
14. Display program
15. Online display

Figure 10. Interface Between ZB-792 and Other CAD Softwares and Data Bases

A user may choose among ZB-792 descriptive language, digital input and editing and interactive graphic editing (with the IGEST software developed) to carry out graphic input and editing based on his needs. A pattern check software (such as JC-81) can be used to automatically verify a mask. In addition, the display and plotting routines in ZB-792 can also be used to do direct checking. If there are mistakes, IGEST can be used to perform interactive modification. If appropriate, standard and common circuit patterns can be stored in a data bank. Elements in the data bank can be extracted from the data base for assembly. The data base management system (such as the CAD data base QCD-1 under development) has a series of operating commands. The pattern of each data base element is written in ZB-792 language to be entered in the data base. As an element is called, its data is stored as a ZB-792 intermediary file on a disk to be accessed by the graphic editing software. A graphic data base based auxiliary graphic design and manufacturing system can better serve the needs of LSI and VLSI.

The authors wish to thank those in the computer room of the Eighth Laboratory of the Institute of Automation of the First Machine Building Ministry for testing ZB-792.

12,553/9599
CSO: 4008/1019

TRANSPORT LEVEL PROTOCOL AND ITS IMPLEMENTATION

Beijing JISUANJI XUEBAO [CHINESE JOURNAL OF COMPUTERS] in Chinese Vol 7 No 1, Jan 1984 pp 31-40

[Article*by Zhang Wenkuan [1728 2429 1401] Institute of Computer Application, Ministry of Nuclear Industry and Cheng Yunyi [4453 0336 1837] Institute of Computing Technology, Chinese Academy of Sciences]

[Text] Abstract: The necessity of introducing transport-level in a distributed computer network was discussed. The functions of the transport-level of RDC network, the service interface commands and the protocol commands were described. The method of implementing the transport-level protocol was also described.

I. Introduction

RDC Net is a distributed computer network and its transport level implementation lies in communication between processes on different computers. When designing the RDC Net transport level protocol, we relied on "Internetwork end-to-end transport level protocol proposal" drafted in IFIP WG6.1. Now, on the basis of this proposal, the international standard of transport level protocol is formulated in the "Open system interconnected reference model" put out by the ISO SC16 group.

In the distributed computer network, communication between one process and another process has very important significance. In this environment, many host computers are connected with communications subnetworks and network users are no longer limited to using the resources on a certain computer but can also access the resources of any of the computers distributed in different geographical locations in that network; the user can even complete an entire task through the cooperation of different host computers in the network. The system operates with the process as the basic unit. Thus, cooperation between different computers qualitatively is cooperation between processes distributed over these host computers. Therefore, it is necessary to implement communication between one process and another process on a network wide basis.

---

*Paper received 23 Nov 82.

To achieve the above goal, when a host computer is connected to a communications network, not only should the data transmission task between any two data terminals (DTE) that the packet level provides be implemented, but the functions of a higher level--the transport level--should be added. Speaking in terms of the levels in the network, the RDC Net packet level is above the packet level, and provides a clear remote communications service applications process.

The network's transport level is made up of many entities called transport stations. The transport level uses the network data transmission service provided by the X.25 packet level and exchanges information between the transport stations, in line with the transport level protocol, providing the transport level user with transport service so that the transport level user need not be concerned about the details of the transmission process. The transport level user can be the user program or applications system, such as "the mini relational database system MRDS in the network environment" or the "remote job entry system RJE" and can also be the end user (using the transport service indirectly through the PAD).

II. Main Functions of the RDC Net Transport Level

The RDC Net transport level provides an addressing method for communications between processes network-wide and supports exchange between processes with the information block as the unit. There are four main functions of the transport level:

•addressing;

•setting up, shutting off and cutting between connections;

•data transmission based on links; and

•flow control and error control.

1. Addressing

The major problem in interprocessing communications is mutual discrimination. Since the processes of both sides in communications can be situated in different locations in a network, they work under different operating systems. Each operating system, has its own command conventions for processes. The concept of the 'information port' [xinkou [0207 0656]] is introduced as the only marker process network-wide. Each transport station is responsible for managing the corresponding relations between the process and the information port within the system.

Correspondence between the process and the information port may be permanent or temporary. Processes which provide service to normal communications through the network may establish permanent correspondences between the process and the information port. These information ports are called well-known information ports; other processes, before they use network communications, must request assignment of an information port and the temporary establishment of a correspondence between the process and the information port. In

the RDC Net transport level, a process can use many information ports, but one information port can only represent one process. Each information port has one information port identifier. It is made up of the DET address of the transporting station and the information port number within the local station.

2. Setting up, shutting off and cutting between connections

The connection between a pair of information ports is called a link, and the identifier of a pair of information ports makes only one link on the whole network. The link is used to support the dialogue between information ports (i.e., between a pair of processes).

Information ports which correspond to shared resources and which also can be shared are called shared information ports. Dialogue with many remote processes can be supported simultaneously on such information ports. Information port C in Figure 1 is a shared information port.

The RDC Net transport level permits links to be established between two different local information ports. Thus, access methods for local resources and remote resources in a net are entirely the same.

The link has a one-to-one correspondence with the virtual circuit provided by the packet level.

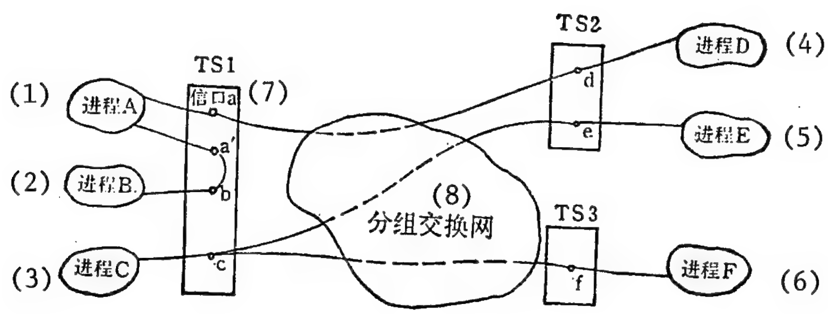


Figure 1. Links

Key:
1. process A
2. process B
3. process C
4. process D
5. process E
6. process F
7. information port a
8. packet exchange net

The possible links are illustrated in Figure 1.

Links may be established between any two information ports in a network. But only when there must be communication between two information ports can the link between them really be established. In terms of implementation, establishing a link in a network is establishing a link item separately for the link between two terminal transport stations and establishing a one-to-one corresponding relationship with the X.25 virtual circuit for this link. After the link has been established, addressing the established link is simplified by using the link index number and does not require global link addresses.

When the data transfer between two information ports has been completed, or for other reasons, the user can request that the link be closed. There are two ways of shutting off a link: normal close and forced close. The normal close mechanism can guarantee that all the user's data can be safely received by the other party before the request to close the link has been transmitted. The forced close mechanism immediately closes all links with the designated information port and closes the information port. This mechanism may lose user data that has not yet been transmitted to the other party.

The RDC Net transport level also provides a link switching function, i.e., after a link has been established, one party can switch this link to another information port (as illustrated in Figure 2). For example, the user may set up a link with the general control process (an information port corresponding to it is a "well-known" information port) of a certain system, and it is responsible for switching the link to corresponding information ports relevant to the processing process in such a way that the processing process can process the user's service request.

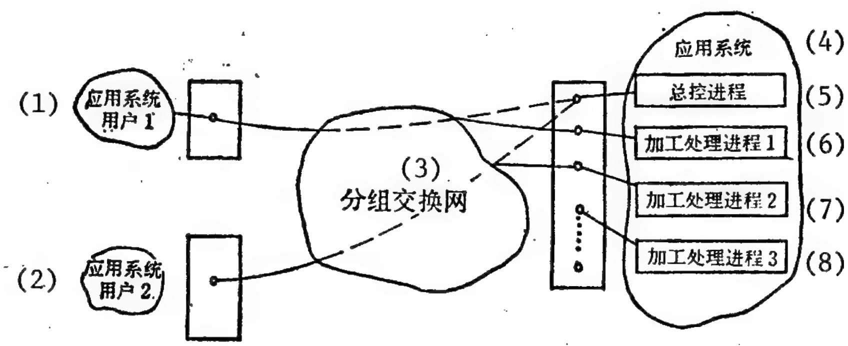


Figure 2. Link switching

Key:

1. applications user 1
2. applications user 2
3. packet exchange network
4. applications system
5. general control process
6. processing process 1
7. processing process 2
8. processing process 3

3. Data transmissions based on the link

After a link has been set up between two information ports, the link is in the data transmission state. From the user's point of view, data transmission carried out over the link is symmetrical and duplex. Duplex can transmit data to both parties simultaneously, exchanged information is divided into letters and telegrams.

A letter is a section of user information of variable length and in implementation its maximum length is 32K bytes. This length permits the user usually to place a section of information which is logically complete, so that the user avoids the complex situation of having to break the information up into sections when transmitting and putting it back together when receiving. Thus, although transmission uses a small data unit--the frame--on the link, from the user's point of view, at the transmitting end the letter is passed in its entirety to the transport station and at the receiving end is relayed in its entirety from the transport station to the user in something like a "MOVE" operation. The mechanism for implementing this function is for the transport station to cut the letter up into a series of successive fragments and pass them to the packet level and send them to a remote transport station. The receiving transport station combines the many fragments received from the packet level into a letter and passes it on to the receiving user process.

A telegram is a section of information 8 bits long. It has priority transmission on the link and is not restricted by flow control.

4. Flow control and error control

To ensure that information flowing into the network does not create crowded conditions on the network and to ensure that the information is received correctly, the RDC Net transport level provides flow control and error control functions.

The flow control mechanism uses the reserve buffer method. That is, it demands that the receiving party first provide a letter reception area before it can receive letters. The transport station records the head address of each letter reception area provided by the user and then sends an 'information material' [xinhuo [0207 6303]] to the other transmitting station over the link. Each information material permits the other party to send a letter. The information material can effectively execute the flow control system on a link and coordinate the synchronicity between the processing capabilities of the sending and receiving parties.

The error control system is primarily a sequential control system. For each link, the transport station manages a letter sending number and receiving number. After the link is established, the two ends obtain initial synchronization, then each letter has a sending number and on the basis of a comparison of the receiving number and the sending number the receiving party checks for drop outs and repetitions in the letter. Letters with sequential errors are to be abandoned, and for letters which pass the sequence check, the receiving party notifies the sending party with the transport level protocol command "letter acknowledged".

III. Transport level service interface commands and transport level protocol commands

The service interface commands and transport level protocol commands were designed on the basis of the RDC Net transport level functions. The service interface commands describe the service characteristics of the transport level and the protocol commands describe the protocol characteristics of the transport level.

The user employs the services provided by the transport level by means of the transport service interface commands. Service interface command parameters include two important pieces of information: interface control information and interface data. Interface control information is information which the user sends to the transport station or the transport station sends to the user to complete the interface action between the user and the transport station. Interface data is data which the user sends to the transport station to communicate with a user on an equal footing at a remote location or it is data from a remote party which the transport station relays to the user for a similar reason.

Protocol command parameters include protocol control information and user data. Protocol control information is information which is sent to coordinate work between corresponding transport stations; user data is data which represents the user processes sent or received.

Table 1 below sets out the categories and functions of service interface commands which the transport level provides to the user, and Table 2 sets out the categories and functions of important transport level protocol commands.

IV. Overview of Implementation

1. Implementation Environment

RDC Net transport level is implemented on a DJS-100 minicomputer service as data terminal equipment. The transport station is implemented through a user process (also called user task) of the RDOS operating system.

The RDC Net queue system has an input queue TSINQ for the transport station. Using the GETS primitive, the transport station can access a processing list from TSINQ. The processing list is either a transport level service interface command sent by the transport level user or an interface command passed by the packet level. After the transport station finishes executing a user command, the response to the user, using the PUT primitive in the processing list form, is put into the user process input queue. When the transport station requests a packet level service, the PUT primitive is used to put the interface command in the packet level process input queue.

The RDC Net buffer management system provides a common buffer pool for transport level, packet level, and link level processes. When a transport station wants to exchange protocol commands with a remote station and prepares information for this, it requests a buffer area from the buffer pool. The buffer area reserves sufficient space for the packet level and link level to insert information, thus avoiding the need to shift data between buffer areas.

Table 1. Transport Level Service Interface Commands

Category		Operational Mnemonic	Major Functions
Service provided by transport level on user demand	Information, link management	OPPT	Opens correspondence between the information port and the information port and setup process
		ALPT	Distributes and opens state correspondence between information port and information port set up process
		CLPT	Closes information port
		LISTN	Ready to receive request from remote party to set up link
		OPLI	Request to set up link with remote information port
		EXAG	Switch link to another information port
		CLLI	Normal link closing
		ABOT	Force closing of all links and the information port at the indicated information port
	Data Transmission	SEND	Send letter
		BUF	Buffer ready for receiving letter and other party notified by information material mode
	Synchronicity	RESET	Reset link already in data transmission stage
		STG	Send telegram
Transport station reports to user		RDATA	Report that letter sent by remote TS received
		RTG	Report telegram sent by remote TS received
		RCLI	Report link cleared
		RREST	Report link reset

Table 2. Transport Level Protocol Commands

Operational Mnemonic	Main Functions
LI-INIT	Request to set up a link or to acknowledge request to set up a link
LI-TERM	Request to terminate a link or acknowledge that a termination of a link
LI-LT	Send letter section
LI-ACK	Acknowledge that letter correctly received
LI-TACK	Acknowledge that telegram correctly received
LI-NACK	Negative response to letter
LI-CRD	Send information material

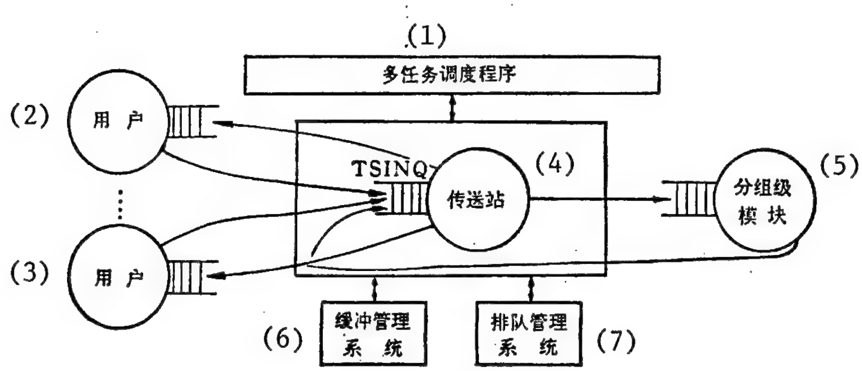


Figure 3. Diagram of transport level implementation

Key:
1. Multi-tasking Transfer Program
2. User
3. User
4. Transport station
5. Packet level module
6. Buffer management system
7. Queue management system

2. Data structure

The transport station contains two important data structures and operates entirely on the basis of these two data structures. They are: information port table and link table. Because the content of these two tables can only be read and revised partly through the operation of the transport station, the interface between the transport level and the packet level and the interface between user processes is very clear.

(1) Information port table

As described above, defining a group of information port in a net and setting up correspondence between an information port and a process in the local area enabling the realization of addressing processes throughout the entire network. The RDC Net's transport stations set up these correspondences by means of the information port table. The correspondence can be permanent or it can be temporary. Information ports which can establish permanent correspondences are called dedicated information ports and information ports which can only set up temporary correspondences are called public information ports. Therefore, the information port table is divided into dedicated information port areas and public information port areas. The content of each information port table is as follows:

Name of process corresponding with this information port;

Opening characteristics;

Allocation characteristics*;

Link counter on this information port;

Listening characteristics; and

Other information.

<Name of process corresponding with this information port> When the system is initialized, the process name corresponding to this information port is entered in this field in the permanent correspondence. In the case of the temporary correspondence, after the transport station has assigned an information port in accordance with user request, the user's process name is entered in this field. These correspondence relationships are maintained in the overall stage of information port activity.

<Opening characteristics> When this information port is in the active stage it is "1".

<Link counter on this information port> Records the number of links established currently with this information port.

<Listening characteristics> When the user process providing service is ready to accept a request from the other party to establish a link it is "1".

<Allocation characteristics> If a public information port has already been allocated to a process, the allocation characteristic is "1".

(2) Link table

When there has to be dialogue between two processes on a net, the transport layer builds a link between the two corresponding information ports. This means building a link table for the transport station at both ends. The content of the link table is as follows:

Local information port identifier;

Remote information port identifier;

Link state;

Maximum length of letter which can be sent;

Number of information material provided by other party;

Receiving letter buffer information;

Sending letter number;

---

*Only the public information port area information port tables have this characteristic.

Receiving letter number;

Process name of local user extraordinary situation processing procedure; and

Other information and work elements.

<Local information port identifier> and <remote information port identifier> are the only markers of this link.

<Link state> refers to whether or not the link is in the data transmission stage. If the link is in the data transmission stage, the state field also indicates whether or not it is awaiting acknowledgement of telegram sent or whether it is awaiting the information material state on this link. If the link is not in the data transmission stage, the state field indicates whether or not this link is awaiting acknowledgement of the other party setting up a link or whether or not it is awaiting acknowledgement that the other party has cleared the link. Depending on the different values in the state field, the transport state takes the appropriate action with regard to the user or packet level request.

<Maximum length of letter which can be sent> is in the link establishment stage and both parties determine the maximum length of the letter through exchange of initialization information.

<Number of information material provided by other party> The party receiving the letter provides a buffer for the letter being received and every buffer which can receive a letter gives the sending party an infomation material. The sending party records the information material number in this field in the link table. When the sending party has an information material, it can send the letter to the other party.

<Receiving letter buffer information> indicates the place where the buffer address provided by the receiving party for receiving the letter is stored.

<Sending letter number> and <receiving letter number> are used to control the sequence of sending and receiving letters.

<Process name of local user extraordinary situation processing procedure> The user would like to be able to process a telegram received as quickly as possible and process quickly a clear or reset report sent by the system, therefore frequently sets up an extraordinary situation process procedure. In the link set up stage, this procedure name is recorded in the link table so that when an extraordinary situation occurs, it can be promptly notified.

We hope that the transport station can serve multiple users simultaneously, and the link table plays an important role in resolving problems of operating at the same time. The transport station receives a user's service request, and after constructing the transport level protocol commands passes it to the packet level for transmission to a remote transport station. At the same time, the current state on the link and relevant comprehensive information is preserved in the corresponding places in the link table. Before a response is

received from the remote transport station, there is often nothing for the local transport station to do on the link so it switches to other user service. When the transport level protocol command of confirmation from the other party arrives, the transport station carries out the processes depending on the various information and the state which has been stored in the link table. It can be seen that through inquiry and revising the link table, the transport station can simultaneously process the various jobs on multiple links without any confusion.

3. General control and rough outline of transport station

The transport station process is activated by a multi-task calling program or by the queue system of the RDC Net when it places a processing list in the transport station's input queue (TSINQ) and checks the queue for control. After the transport station is activated, it accesses the processing list in order from its input queue. The general control program of the transport station first of all analyzes this processing list to determine whether it is a service request provided by the user (i.e., interface commands sent by the user to the transport station) or interface command sent by the packet level. If in the packet level interface command it indicates that it is a data packet received from a remote party, the general control program determines which class of transport level protocol commands is contained in the data packet. Then, the general control program analyzes whether or not the operational code of these interface commands or transport level protocol commands is rational. If irrational, it carries out error processing, and if rational transfers control authority to the user interface command processing program and the packet level command processing program or the transport level protocol command processing program.

The transport level general control is outlined in Figure 4.

4. Important functions of processing programs

(1) Functions of user interface command processing program

*Checks to see if user issuing interface command to transport station has the right to use the information port or link indicated in the command parameters. If the user does not have that right it is notified, and the transport station cannot provide it service.

*Establishes, uses and revises link table content, especially the states of corresponding links.

*Carries out different processes on the basis of the different operational codes of the user commands. With regard to operations which are not related to network activity (such as information port operations), when command processing is completed, it responds to the user process requesting service. With regard to service requests which are closely related to network activity (such as opening and closing links, sending letters and telegrams) the transport station constructs a packet of the user data containing the transport level protocol commands and requests the packet level to send it to the remote party using the corresponding interface commands. Then it waits for

confirmation from the remote transport station on the corresponding link. There is another type of user interface command (such as switching and forced closure commands) for which after the transport station has carried out its processes, it requests the packet level to send information to the remote party and without waiting for confirmation from the other party it immediately responds to the user.

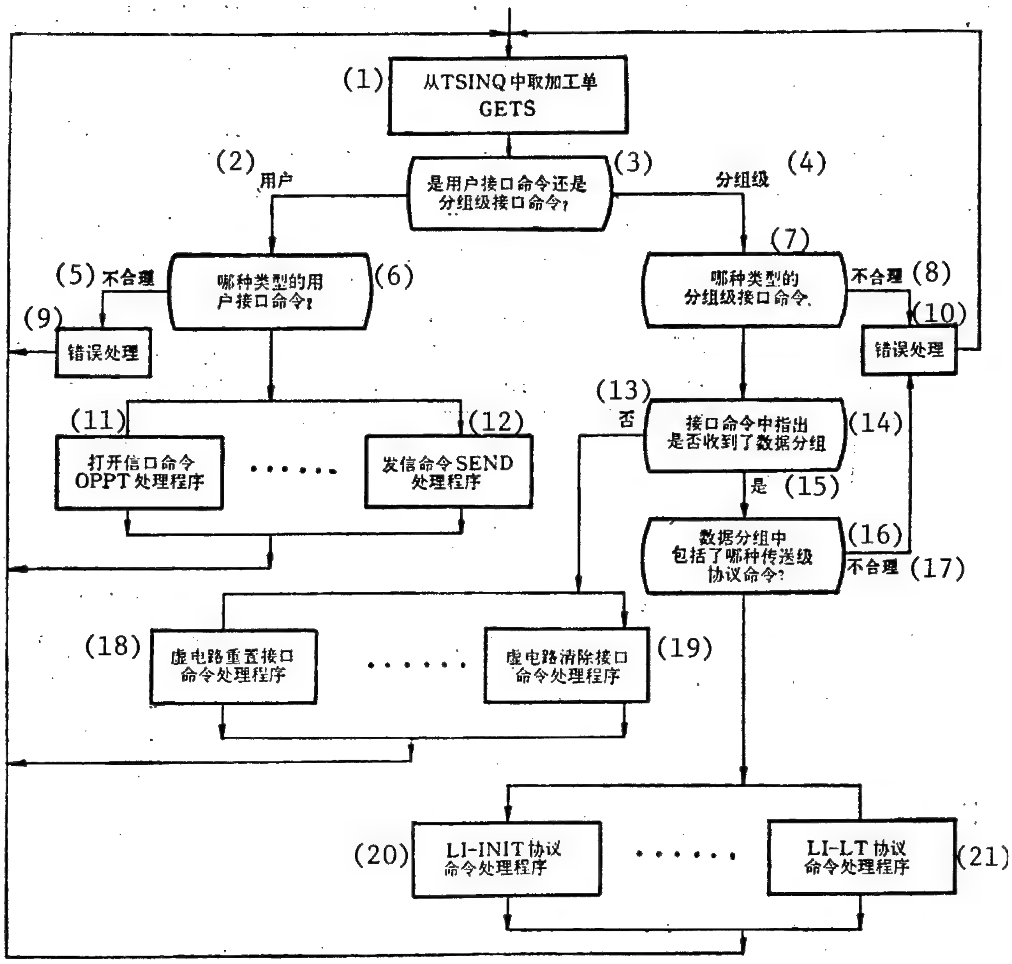


Figure 4. Diagram of general control of transport station

Key:

1. Access processing list GETS from TSINQ
2. User
3. User interface command or packet level interface command?
4. Packet level
5. Irrational
6. Which type of user interface command?

[Key continued on following page]

7. Which type of packet level interface command
8. Irrational
9. Error processing
10. Error processing
11. Open information port command OPPT processing
12. Send letter command SEND processing
13. No
14. Is data packet received indicated in interface command?
15. Yes
16. Which transport level protocol command included in packet level?
17. Irrational
18. Virtual circuit reset interface command processing program
19. Virtual circuit clear interface command processing program
20. LI-INIT protocol command processing program
21. LI-LT protocol command processing program

(2) Functions of transport level protocol command processing program

•Checks whether or not the packet received on the indicated link under the present state is rational and if not transfers to error processing.

•Uses and revises link table items (especially, link state).

•If the transport level protocol command received is a request confirmation, the transport station constructs a protocol command with confirmation and passes it to the packet level to be sent to the other party.

•If the transport level protocol commands received are confirmation information sent by the other party's transport station, the transport station responds to the service request of the user on the same link.

•If the letter segments received from the packet level have been put together into a letter, the transport level notifies the user that there is a letter that has been received on this link. If a telegram sent by a remote party is received from the packet level, the transport station notifies the user itself.

(3) Packet level interface command processing program

In addition to processing data packets containing transport level protocol commands received from the packet level, the transport station also processes other interface commands received from the packet level, such as packet level reports of clearing and resetting virtual circuits. Then the transport level closes or resets the corresponding link and reports to the user what type of extraordinary situation occurred.

5. Examples

We will explain the processing procedure in the example of sending a letter and its relevant commands. The transport station receives a user's request to send a letter. First of all it checks to see whether or not the user has the right to use this link, whether or not the link is in the data transmission

state, whether or not the other party has provided an information material, whether or not the letter length exceeds the permissible length, etc. When these conditions are not satisfied, the local transport station responds to the user with the reasons why it cannot be transmitted. Otherwise, the letter is broken up into a series of packets which are passed to the packet level and sent to the other party's transport station. For this a series of buffers is requested and the data in the user's letter is moved into these buffers section by section. Then the letter sending number is revised and such information as the count of information material provided by the other party is revised. Finally, the link state and corresponding information is recorded in the link table and it turns to other user service. When the transport station receives protocol command of letter confirmation from the other party's station, showing that the entire letter has been received properly, it informs the user that the letter has been sent.

If the user requests that a letter be sent and has passed the rational check but the other party has not yet provided an information material, after the transport station sets up a link lacking an information material state, the user's request to send a letter is registered and it switches to other user service. When information material provided by the other party arrives, the process of sending the letter continues again.

Figure 5 is a diagram of the work of the various modules involved in sending a letter.

V. Considerations for further development

Data transmission between two information ports can still be carried out in letter or telegram mode. In gearing to transaction processing applications, the letter and telegram mode is frequently highly efficient. When the RDC Net packet level implements data report service, we can implement letter and telegram service on this basis.

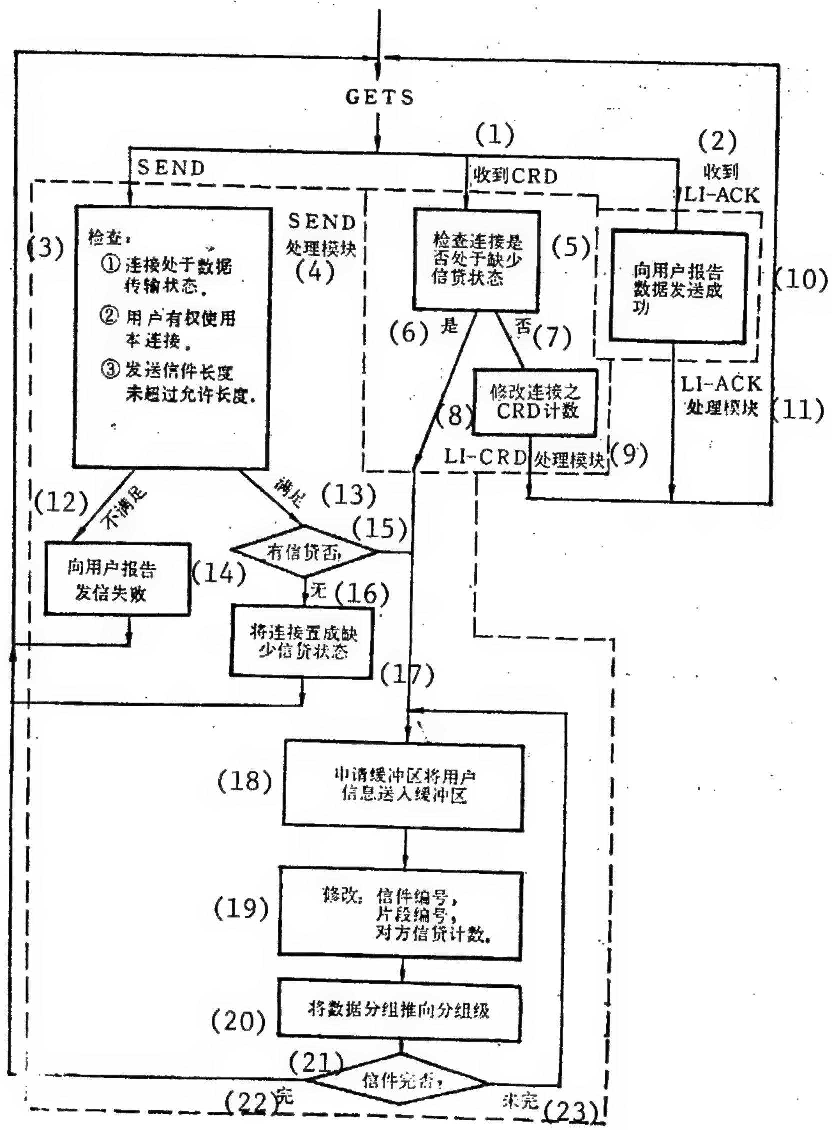


Figure 5. Diagram of the Work of the Various Modules Involved in Sending a Letter

Key:

1. Receive CRD
2. Receive LI-ACK
3. Check
   (1) link in data transmission state
   (2) user has right to use link
   (3) sending letter does not exceed permissible length
4. SEND processing module

[Key continued on following page]

5. Check whether link is in no information material state
6. Yes
7. No
8. Revise link's CRD count
9. LI-CRD processing module
10. Report to user that data has been sent
11. LI-ACK processing module
12. Not satisfied
13. Satisfied
14. Report to user letter transmission failed
15. Is there an information material?
16. No
17. Set link to no information material state
18. Request buffer area to send user information to buffer area
19. Revise:
    letter number
    segment number
    other party information material number
20. Push data packet to packet level
21. Letter finished?
22. Finished
23. Not finished

8226/12712
CSO: 4008/244

TERMINAL ACCESS, PACKET ASSEMBLE AND DISASSEMBLE (PAD)

Beijing JISUANJI XUEBAO [CHINESE JOURNAL OF COMPUTERS] in Chinese No 1, 1984 pp 41-46

[Article*by Wu Zhimei [0702 1807 5019], Xu Guirong [1776 2710 2837], Sun Simin [1327 0934 2734], and Song Huangang [2646 3562 0474], Institute of Computing Technology, Chinese Academy of Sciences]

[Text] I. Introduction

The RDC Net is an experimental computer network with the primary goal of resource sharing. Its software has five levels. PAD belongs to the meeting and expression levels and is one of the important parts of the network communications system. The RDC Net PAD was designed in accordance with CCITT proposals X.28 and X.3, taking into account the specific environment of the software of transport stations which make up the RDC Net nodes and to support three types of terminal equipment: local five-bit telegraphic and local seven-bit telegraphic terminals and remote seven-bit terminals on the intercity telephone network (see Figure 1). The basic goal of PAD is to make it easy for the various types of terminals to call the service programs on the network.

II. Terminal calling network resources

A terminal connected to any node can call any service program on the network. If this service program is willing to serve it, it can share this resource. The terminal user need only know the service program's code name and it can provide this service without the need to know the geographical location of the service program. The terminal user can view the entire network as an enormous mainframe computer with all the service programs waiting for him to call them and ready at any time to work for him (see Figure 2).

Information is transmitted between nodes in the RDC Net in frame format and transmission speed is very fast. Information is transmitted between the terminal and the node in character mode and the speed of transmission is determined by the equipment. The means by which the terminal user calls the service program he wants at any time from a remote location is the result of the common efforts of the communications system in the network in which the roles played by PAD are mainly the following:

*Paper received 23 Nov 82.

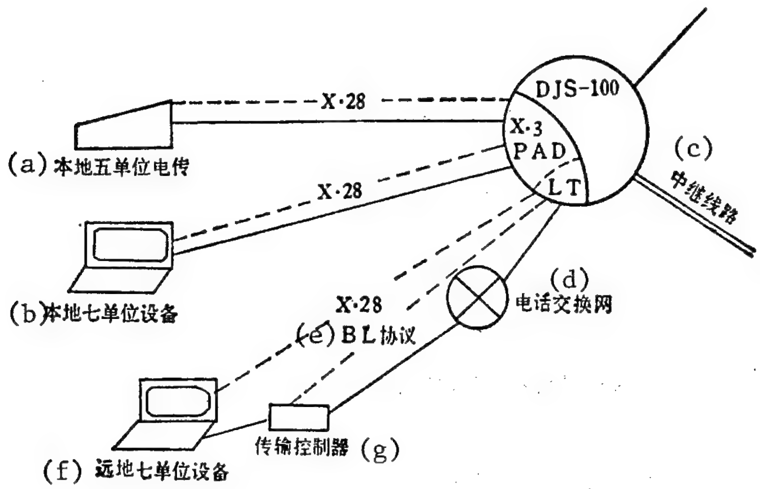


Figure 1. RDC Net PAD

Key:
- a. local five-bit telegraphy
- b. local seven-bit equipment
- c. trunk line
- d. telephone network
- e. BL protocol
- f. remote seven-bit equipment
- g. transmission controller

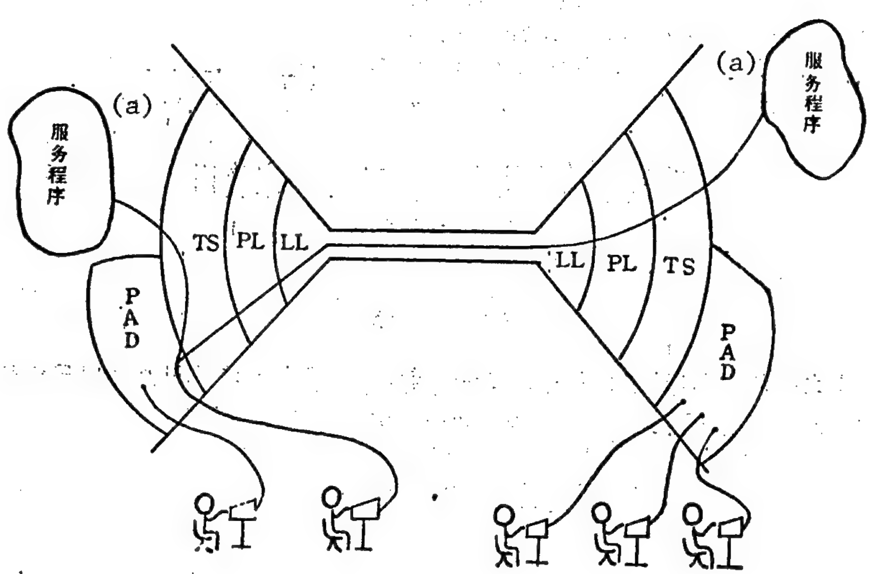


Figure 2. Terminal calling service program

Key: a. service program

(1) Collecting the character information sent to the network by the terminal and putting it in packet form for transport between modes; breaking the packet information sent from the network to the terminal into characters and sending it to the terminal. Linking up the character flow between the terminal and the network to the packet flow within the network.

(2) Providing a buffer for the character flow which the terminal sends to the network and the packet flow which the network sends to the terminal, preventing excessive terminal data from innundating the network and preventing data sent from the network to the terminal from assaulting the terminal at too high a speed, and although crossing two different circuit media, the flow of information going back and forth between the terminal and the remote service program can be maintained in an orderly state after going through the PAD information buffer.

(3) Turning on and turning off connections between the terminal and the service program for the terminal user.

(4) Expressing information provided on the terminal according to the terminal characteristics, such as whether or not a space need be added after a carriage return or a line.

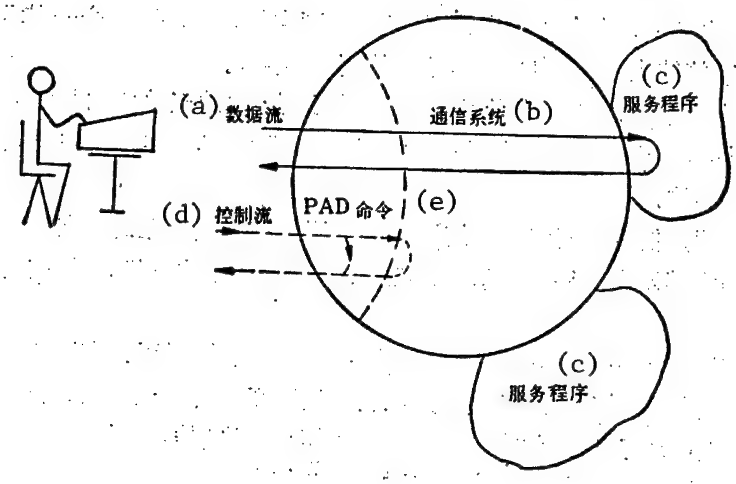


Figure 3. Data flow and control flow

Key:

a. data flow
b. communications system
c. service program
d. control flow
e. PAD command

The five-bit telegraphic character set includes only 53 items, but with the addition of a function key after the original characters the size of the original character set can be doubled and in addition to distinguishing between upper and lower case, it has all the functions of the standard ASCII character set. Users working on five-bit telegraphic equipment can view this telegraphic equipment as seven-bit equipment with the usual ASCII character set, except that those low use ASCII symbols must be replaced by two telegraphic characters. Turning the characters input by five-bit telegraphic equipment into ASCII characters and then converting the ASCII characters output to the terminal into five-bit telegraphic characters is implemented through an interrupt program. The PAD program cannot tell the difference between these character sets.

Currently, the interference signals on China's public telephone network are considerable and the digital communications error rate in communicating directly on such a network is very high, being especially severe in the municipal telephone networks of such large cities as Beijing and Shanghai. In the RDC Net, a transmission controller is added to terminals for remote terminals on the intercity telephone network, a terminal link management process LT is added to the node computer and a basic data communications protocol BL is implemented between transmission controller and LT (see Figure 1) to eliminate interference signals on the municipal telephone network and to ensure the reliability of data signals transmitted on the municipal telephone network circuits. Users working on remote terminals use an ordinary telephone to dial up a node and after receiving the answer tone sent back from the computer can call service programs on the RDC Net just like local terminals. As far as the computer is concerned, information the LT process passes to PAD are characters sent from a remote terminal since PAD cannot tell the difference between remote terminals and local terminals.

In short, no matter what kind of terminal user it is, it thinks it is a piece of ASCII character equipment directly calling a service program on the network and PAD always provides certain auxiliary services for ASCII terminal users.

III. Data flow and control flow

Information flow between terminal users and the network can be divided into two classes: one class is data flow and the other is control flow.

Data flow is used for dialogue between the terminal user and the service program. PAD does not analyze this type of information, but merely stores and relays it.

Control flow provides auxiliary services for the dialogue between the terminal user and the service program. Before the dialogue begins, a link must be set up between the user and the service program, and when the dialogue is over, the link that has been set up must be closed, and during the dialogue process, information which the user might request be sent to the terminal observes certain formats, this type of auxiliary information flow is called control flow (see Figure 3). PAD analyzes the process, and sends the results to the user. In control flow, the information that the terminal passes to PAD is called PAD commands, and information that PAD passes to the terminal is called the PAD response.

So that it can discriminate PAD commands and user data, it has been decided that in the RDC Net, PAD commands always start with an "esc" character, and not until concluded by a carriage return character is an "esc" character permitted in the user data.

The character echo of the terminal user in the RDC Net comes from the local, the echo of the local five-bit and seven-bit equipment comes from the node computer, and the echo of the remote terminal comes from the transmission controller. On the basis of the echo information, the user can be confident that the information passed to the network is correct.

To avoid having the echo information and the information passed by the network to the terminal becoming confused, while the network is transmitting information to the terminal it is locked in terminal to network information flow. Under normal circumstances, the terminal and the network work in dialogue mode.

Under certain circumstances, the terminal user can break off this dialogue process at any time. When the network is transmitting information to the terminal, PAD does not accept ordinary character input from the terminal, but still monitors halt and break signals sent by the terminal. When the terminal user discovers that a temporary halt signal must be sent to halt terminal output because the printer has run out of paper or for some other reason, it is only necessary for a temporary halt signal to be sent to the network. When the user wishes to resume output, it requires only that any character be typed to resume. When it becomes necessary for some reason for the terminal user to stop output to the terminal, it is only necessary to send an interrupt signal to the network and the system is restored at that time to the state when it was just linked to the service program.

IV. PAD and user interface

Interface between the PAD and the terminal user observed CCITT X.28 proposal (see Figure 1). An ordinary terminal user call for a service program includes the following steps:

(1) Establishing a link with the service program using PAD commands.

(2) Dialogue with the service program using the service items provided by the service program. At this time, the terminal user can insert certain dialogue with PAD as needed.

(3) Breaking off an established link by use of PAD commands.

Dialogue between the terminal user and the service program is an ASCII string except for the "esc" character. Commands used in terminal user-PAD dialogue are set out in Table 1. The content of these commands is as defined in CCITT X.28 proposal.

Table 1. PAD Commands

Name	Form	Function
Parameter initialization	Ⓔ PRO CR	Sets PAD parameters at initial value given by system
Read parameter	Ⓔ PAR? parameter #1, parameter #2,...CR	Reads PAD parameters, if parameter # expression is not given, reads entire parameter
Write parameter	Ⓔ Set: parameter #1: parameter value 1; parameter #1: parameter value 2..., CR	Writes parameter value indicated by user
Write and read parameters	Ⓔ Set? parameter #1: parameter value 1; parameter #2: parameter value 2, ..., CR	Writes parameter value indicated by user then reads it out
Request link	Ⓔ CON, service program code word CR	Sets up link with service program
Request set 0	Ⓔ RESET CR	Sets virtual circuit on link to 0
Request interrupt	Ⓔ INT, interrupt code CR	Sends special data of character, not restricted by packet network flow
Request close	Ⓔ CLEAR CR	Break off established link
Test link	Ⓔ STAT CR	Enquires about status of current link

V. Interface between PAD and transport station (TS)

There are currently still no standards that can be observed in the interface between PAD and the TS. There are two groups of nearly symmetrical commands in the interface between RDC Net PAD and TS (see Table 2).

The table lacks a command to pass link set up from TS to PAD because at present RDC Net still does not permit terminals to be the object of calls, thus there is no need for such a command. If a service program that automatically calls the terminal or for a terminal to call a terminal ever appears, this command will be added.

VI. Working process of the PAD procedure

In the RDC Net, PAD is a procedure. It processes the commands from many terminals and TS. The commands used by terminals and TS for dialogue with PAD in a brief time are completely random. For certain commands, the responses can be given without going through PAD processing but each terminal wants to obtain response information as quickly as possible. This is a problem which faces the PAD process.

Table 2. Interface commands between PAD and TS

Name	Form PAD TS	(TS PAD)	Function
Set up link	OP-LI m1, m2		Request setting up link between information port m1 and information port m2
Close link	CL-LI n		Request close link #n
		R-CL-LI n	Remote request to close link #n
Set 0	RESET n		Request setting 0 on link #n
		R-RESET n	Remote request to set 0 on link #n
Send data	CSEND-LI n, ad		Request sending a packet on link #n packet address ad.
		R-CSEND-LI n, ad	Packet sent from remote location on link #n, packet address ad.
Send telegram	SEND-TG n, X		Request sending a telegram on link #n telegram code is X.
		R-SEND-TG n, X	Telegram sent from remote location on link #n, telegram code is X.

To resolve this contradiction, the PAD process adopts two measures. First, for each terminal a terminal characteristics table is set up recording the PAD parameters of each terminal and such information as the information port number and link number. PAD commands for writing PAD parameters and inquiring about the link state need only inquire of the terminal characteristics table according to the circuit number once to be able to give the terminal response information.

Second, those commands which cannot be immediately responded to by going through PAD processing once, such as link commands, are transferred to TS commands of links after being processed by PAD once, and after TS processing they are answered, then PAD carries out a second processing and the response information is output to the user. It takes a long time for a response from the TS to link commands referred to TS, and in this period of time, PAD is not just waiting for the TS response, but continues to process the other commands which are collecting in it. In this way it is possible for there to be many commands for a brief period referred to TS without waiting for an answer. To avoid confusion of TS response information, an information section is appended indicating to which terminal the command belongs. When the TS gives a response, this information section is returned intact and when the PAD processes this command for the second time, it knows to which terminal the response should be given.

In this way the above mentioned contradiction is resolved and a response is sent to the terminal in real time under the fastest possible conditions.

VII. Conclusion

Currently, PAD does not have the ability to provide service programs or terminal calling terminal, and in the future when these demands are presented, these capabilities will be added with very small revisions in PAD.

PAD is something which comes in direct contact with the terminal equipment and the user, there are many types of terminal equipment, user demands are very broad, and our knowledge and experience are very limited, therefore in design there are certainly many deficiencies and shortcomings and these we are prepared to constantly improve and revise in use.

8226/12712
CSO: 4008/244

MINI-RELATIONAL DATABASE SYSTEM (MRDS) FOR DJS-100 SERIES

Beijing JISUANJI XUEBAO [CHINESE JOURNAL OF COMPUTERS] in Chinese Vol 7 No 1, Jan 1984 pp 47-58

[Article by Xing Junying [6717 0913 5391], Graduate School, Chinese Academy of Sciences, Zheng Yuanxin [6774 6678 2450], State Instrument General Bureau, Li Kai [2621 0418], Computer Center, State Planning Commission, Zhou Mingtao [0719 2494 7118] and Wang Haigeng [3076 3189 5087] Institute of Computing Technology, Chinese Academy of Sciences]

[Text] Overview

With the development of computer networks, the problem of distributed data processing of necessity arises. The Mini-Relational Database System (MRDS) is an experimental system which was developed for this reason. MRDS was designed for the DJS-100 series computer and works under the support of this computer's realtime disk operating system (RDOS). Data with the base as the unit is distributed on the node computers in the RDC Net and when equipped with a network communications control system on the node computers, the many users on the network can simultaneously query the database for information. MRDS also provides the database administrator with such functions as convenient library creation, and adding, deleting, and revising data. Figure 1 illustrates the distribution of the MRDS system.

The design targets of MRDS were miniaturization, multi-function, and multiuser. It not only has the basic functions of ordinary database systems, but also can satisfy the query demands of general small-scale information search systems, i.e., the user can search the documents by names and keywords. MRDS has a non-procedural search language which takes optimization of query processing, such as inverted index file queries, into consideration in implementation.

MRDS database management system uses the relational pattern. This is because the relational pattern is characterized by a data logic structure that is clear, a query language that is non-procedural, and is convenient for users on the network to use. On the other hand, because the data transmission rate between node computers on the network is several tens to several hundreds times slower than the access speed of the host computer and disk, the primary shortcoming of the relational pattern is the low query efficiency and it does not standout on the network.

The rest of this article will discuss the MRDS query language, and network online computer search subsystem, the database management subsystem and the general implementation of the index functions.

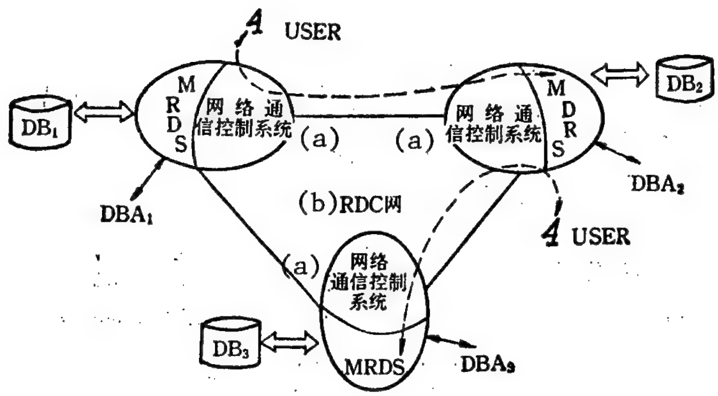


Figure 1. Diagram of MRDS System Distribution

Key:

a. network communications control system
b. RDC Net

II. Query Language SRQL

The MRDS Simple Relational Query Language (SRQL) is a query language that is non-procedural and easy to use. SEQUEL 2 was the primary point of reference when it was being designed. It is geared to two types of applications: one is data processing, such as planning statistics, warehouse management, and banking; the other is information searching, such as document searches and filing. For information searches a data classification pointer was introduced into the query language. SRQL supports four types of data classifications: integers, real numbers, character strings, and pointer types.

We will first cite some examples of the types and characteristics of the query language.

Example 1. Given two relationships:

DEPT (DNO, DNAME, LOC)

EMP (NAME, DNO, BYEAR)

in which all attributes are character strings except for DNO (department number) and BYEAR (year of birth) which are integers. To find the names, ages, and departments of all personnel born before 1921 who are working in departments subordinate to Peking, the following statement may be used:

VIEW R = SELECT EMP*DEPT (NAME, 1981-BYEAR, DNAME)

WHERE EMP. DNO = DEPT. DNO

AND DEPT. LOC = 'PEKING'

AND EMP. BYEAR $\leq$ 1921;

In this example, R is the temporary relation established through the query and is used to store the result of the query. Once it has been established, it can be used in later queries just as relations in general. The user can use the PRINT statement to output any one resultant relation in the database on a printer, local terminal or remote terminal. When the user no longer needs this temporary relationship, it can be deleted with the statement DROPV R.

Example 2. Given one relationship:

PAPER (AUTHOR, TITLE, DATE, MAG, PAGE)

in which AUTHOR and MAG are character strings, TITLE is a pointer, and DATE and PAGE are integers. To find all the articles written after 1975 by S.B. Yao, containing the relevant keyword DATA in the title and published after 1978, and then get the articles' author names, article titles, and places and dates of publication, the following command query is used:

VIEW R = SELECT PAPER (AUTHOR, TITLE, MAG, DATE)

WHERE DATE $\geq$ 1978

AND TITLE = :DATA

OR DATE $\geq$ 1975

AND AUTHOR = S.B. YAO

in which the meaning of =: is the same as above.

The query in example 2 only involves one relation so we call this a single query, while the query in example 1 involves two relations so we call this a double query.

From examples 1 and 2 it can be seen that the query contains three relational operations: adaptive select, project, and join. A query statement is divided into two parts: the SELECT phrase and the WHERE phrase. The SELECT phrase is ranked in the query projection table and the projection table can also rank the attribute names one wishes to get as well as the form of expression (as in example 1). The WHERE phrase gives the element group of the relevant query relationship in the form of a Boolean function extract, generally in the form:

$F_1 V F_2 V \ldots V F_s$

in which F_i is predicate hexi [0678 2649] form ($i = 1, \ldots, s$), i.e.,

$F_i = C_{i1} \wedge C_{i2} \wedge \ldots \wedge C_{im}$

Here, C_{ij} is the predicate ($j = 1, \ldots, m_i$), C_{ij} form as a b, a,b can be expressions made up of an attribute name and constant and $\theta \varepsilon \{=, \neq, >, \geq, \leq, :=. =:\}$.

Table 1 gives the primary statements in SRQL and a simple explanation of them.

Table 1. Primary statements of SRQL

CREATE <relational name><attribute explanation>	create relational pattern
INDEX <index explanation>	sets up index inverted file
LOAD <relationship name><device number>	load relational data
PRINT <relational name><device number> <format explanation>	output result relation
VIEW <relational name>	set up relational view
SELECT...WHERE...	query statement
DROPV <relational name>	delete relation
DELECT...	delete data
INSERT...	add data
UPDATA...	update data

III. MRDS Network Online Search Subsystem

Since the DJS-100 series computers are small and have limited memory, we divided the MRDS functions into two large groups and designed two subsystems around them: the network online query subsystem and the local database management system. The network online query subsystem is designed for network users, its function is to provide the many users on the network with simultaneous query function; the local database management system was designed for database administrators (DBA) to help them maintain and manage the use of the database, such as setting up relations, establishing inverted indexes, and adding, deleting and revising data. Below we will introduce the structure and functions of these two subsystems.

This section deals with the online query subsystem.

1. The structure and functions of the network online query subsystem

To provide data query functions in a network environment it is necessary to deal with the question of network communications. Viewed from the angle of network protocol, there is an applications level protocol between the MRDS and the user. It provides direct information exchange between the user and the system on the lower levels of the communication protocol. Figure 2 gives the network environment in which the MRDS online query subsystem is situated and the interface relationships between it and the user, RDC network communications control system and the RDOS system.

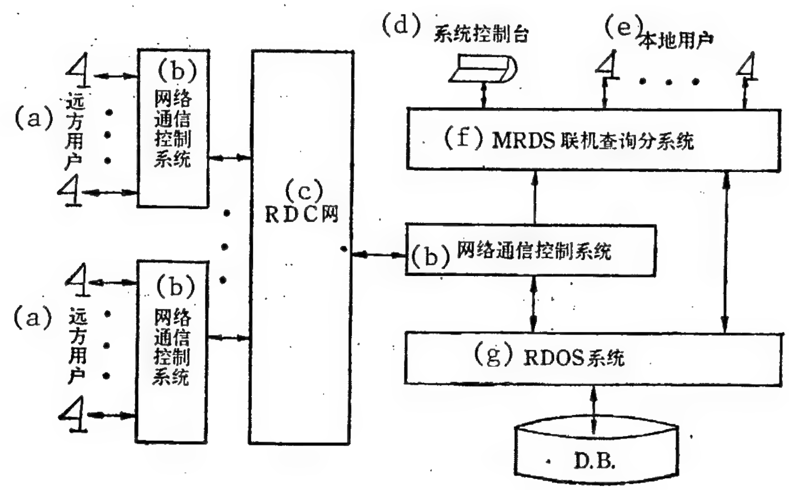


Figure 2. Environment and Interfaces on Online Query Subsystem

Key:

a. remote user
b. network communications control system
c. RDC Net
d. system controller
e. local user
f. MRDS online query subsystem
g. RDOS system

The online query subsystem uses the multi-task management, file management, storage administration and device management functions supplied by RDOS through system call commands and function call commands; it carries out such functions as linking with network users and sending and receiving information on the network through the commands supplied by the network communication control system. The online query subsystem provides a group of commands for multiple users to use the system simultaneously, such as request to use the system, communicating user name, issuing various query commands, and concluding query; it also provides a group of control commands for the database system administrator so that he can monitor and control MRDS operations through the system control desk, such as starting, query, interrupt, switching, and concluding MRDS.

Figure 3 provides a diagram of the online query subsystem structure. The system is made up of six processes, all of which were written as RDOS user processes and operate in parallel under multi-task scheduling on the basis of the priority level to which each has been assigned. Between these processes and between the communication control system processes with them, the primitives for accessing and registering processing lists which have been adopted communicate with each other, and work in uniform coordination, the online query providing convenience for the many parallel user queries on the network, making the system response speed as fast as possible and reducing the time delays created by remote communications.

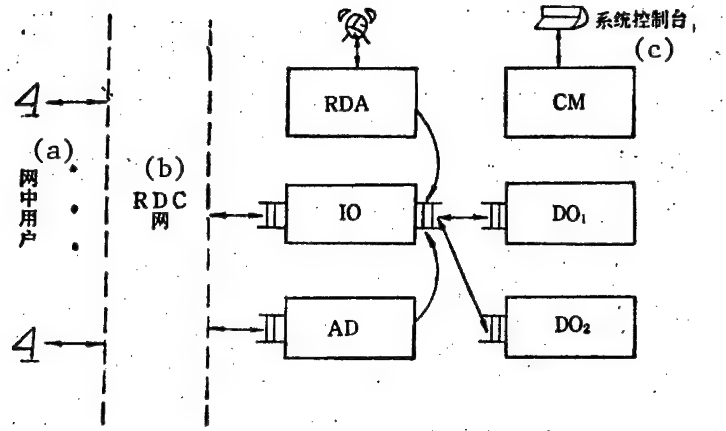


Figure 3. Diagram of the Structure of the Online Query Subsystem

Key:

a. network user
b. RDC Net
c. system control console

When the network user must query the database information through the network, first he calls the MRDS online query subsystem on the local network communications control system, the admissions process (AD) provides admission, allocates a buffer area, checks the user's status, then the query can begin. In the query stage, the IO process sends information in the letter mode and carries out input and output of query results and multiple user commands. The system also provides extraordinary processing functions, such as: when the user must temporarily halt the query or if trouble occurs in communication, the telegram mode can be used to interrupt the system; the system has a letter transmission timer and letter retransmit mechanism to ensure the reliability of letter transmission and prevent "deadlock" of the system. The RDC clock process is implemented by this mechanism. CM is the system control console process and monitors and controls the operation of the entire online query system. DO_1 and DO_2 execution processes are the nucleus of the online query and will be discussed further below.

2. Syntactic analysis

The nucleus of online query is syntactic analysis and query processing, as illustrated in Figure 4.

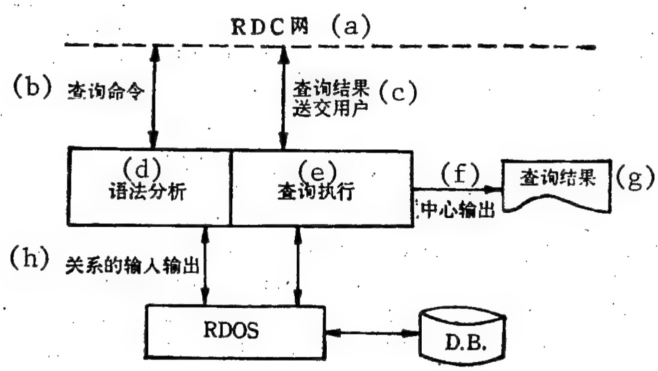


Figure 4. Nucleus of Online Query

Key:

a. RDC Net
b. query command
c. query result sent to user
d. syntactic analysis
e. query execution
f. central output
g. query results
h. relational input/output

After the execution process receives a query statement, the syntactic analyzer carries out syntactic and semantic check of this query statement and gives several error information tables for the errors which violate the linguistic definitions, and does preparatory work for query optimization, finally produces a target code table made up of projection table, Boolean condition tree and relevant forms of expression and sends it to the query execution portion. The syntactic analysis process of complex query condition statements is an follows:

$$R_T(C_1,\ldots,C_m)=\Pi_{U_T}\sigma_F(R_{s_1}|\ X\ |R_{s_2}) \qquad (1)$$

(1) is the formal expression of a query statement: R_T is the result relational name, Π is the projection operational symbol, $U_T = \{E_1, \ldots, E_m\}$ is the projection expression set; σ is the adaptive select operational symbol, F is made up of single order predicate Boolean expression; $|\ X\ |$ is a two element connective, $R_{s_1}|\ X\ |R_{s_2}$ expresses that a certain attribute link will be carried out on the two relations, S_1 and S_2.

To reduce the number of operations and the cost of accessing the disk in the query process as much as possible, Boolean factors of different types should be analyzed and adaptive select be done first, then the link, and finally the projection operation, thus (1) is changed to:

$$R_T(C_1,\ldots,C_m)=\Pi_{U_T}(\sigma_{F_{E_1}}(\sigma_{F_{I_1}}(R_{s1}))|\ X\ |\sigma_{F_{E_2}}(\sigma_{F_{I_2}}(R_{s2})) \qquad (2)$$

in which: F_{I_1} and F_{I_2} mean that the index can be used for adaptive select of the attributes in two relations, F_{E_1} and F_{E_2} mean that the index is not used for adaptive select. Equation (2) may use a five element group expression (F_{I_1}, F_{E_1}, F_{I_2}, F_{E_2},θ), each element has a similar Boolean factor string, is R_{S_1} and R_{S_2} carrying out the Boolean factor string of a link operation, and in implementation, those which belong to the F_I category should be determined on the basis of whether or not this attribute is in the index file and whether or not it should be used. The principle is: when the query condition is a Boolean factor conjunction, if there is an index for an attribute involving a factor, this index should be used; but when the query condition is a Boolean factor extraction, if there are indices for all attributes involving Boolean factors then these indices should be used, otherwise sequential scanning should be adopted.

3. Query processing and optimizing

Since SRQL is non-procedural, the user need not know the physical storage form of element groups in the database nor need the user know how to find the information he wants, but he pays a price in loss of efficiency, therefore optimization of query processing is the key to improving efficiency. Work on query optimization currently can be divided into two types: one type is query optimization through relational algebraic expression transfer; the other is through improving the efficiency of basic operations and adjusting the sequence of executing basic operations so that the estimated value of query processing cost is minimal. However, little work has been done on how to process and optimize queries when the internal space is limited, and we have only done some experimenting.

The relations of the MRDS are stored according to the RDOS disk files. There are four methods of scanning relations: sequential scanning and scanning by index, in which the access unit is the page. Generally, the cost of accessing a record on disk is much greater than the cost of access internally, therefore, when estimating the cost of processing a query, we take the number of pages transmitted internally and externally as the unit. The cost of processing a query is:

$$C(\theta)=C_{IN}+C_{OUT}+C_R\cdot W$$

in which, C_{IN} is the number of pages input; C_{OUT} is the number of pages output; C_R is the time taken for processing; W is the factor of converting the time into number of pages transferred internally and externally. Clearly, the first two items play a dominant role and our goal in query processing optimization is to make the value of $C_{IN} + C_{OUT}$ as small as possible. Below we discuss query optimization processing of single elements and double elements.

The strategy of single element query optimization is:

1) Specialized search relations that are not for project operations require only that the project operation be placed before the operation entering the element group to be realized.

2) When carrying out a project operation, the four functions MAX, MIN, AVG, and SUM are calculated for the digital attributes to facilitate subsequent direct use.

3) The aim of carrying out optimization of Boolean functions during adaptive select is to reduce the time spent processing each element group. First of all, some considerations are made on the structure of the query conditions tree to facilitate the search. The structure of the tree is illustrated in Figure 5. The C_{ij} $(i=1,...,s,j=1,...,m_i)$ in the illustration is the predicate and the dotted line part in the parentheses is the predicate conjunction. The character string comparison operators := and =: should be placed on the right side of predicate operations which take a long time to execute because if the predicate on the left hand side in the conjunction is false, it will save having to execute the predicate operation on the right hand side.

4) In the system, index scans are used for processing queries which can use the indexes.

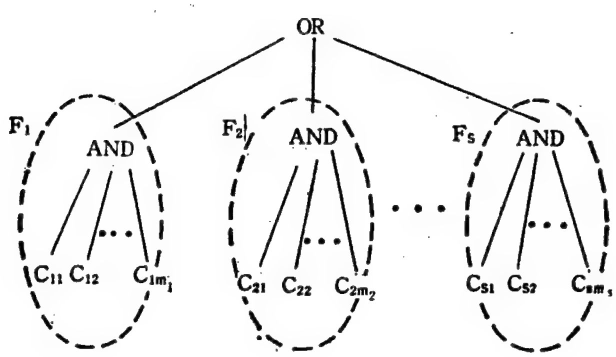


Figure 5. Query Condition Tree

If the third canonical form is satisfied in the relational database, the user can normally carry out the link operation when querying. Improving the efficiency of the link operation is a very critical problem in implementing a relational database management system. Evidence demonstrates that any one multiple element query can be implemented by converting it into a series of two element queries. For this reason, only a two element link operation is provided in MRDS, i.e., a link operation of two relations and this can be implemented in internal space in a very short time.

When a general link operation is executed for each conjunction of a Boolean extract canonical form, since the limiting condition of processing is a predicate conjunction, a single element predicate can be extracted to be processed first and the index scan can be fully utilized. This is a common single element query extract and it can greatly reduce the number of relational element groups in a link. This method has been proven to be one of the most advantageous optimization methods. The query in example 1, Section Two, would be converted into three queries:

i) VIEW E = SELECT EMP (NAME, DNO, BYEAR)

WHERE BYEAR $\leq$ 1921

ii) VIEW D = SELECT DEPT (DNO, DNAME)

WHERE LOC = 'PEKING'

iii) VIEW R = SELECT E*D (NAME, 1981-BYEAR, DNAME)

WHERE E.DNO = D.DNO

If the results E and D derived for EMP and DEPT through i) and ii) after single element adaptive select are very small, then the cost when processing iii) may be much less then not extracting a single element predicate, and the total cost of the two queries would be comparatively less.

Another point is to rationally allocate the buffers used when the link operation is carried out. We divide the buffers we can use into two parts: A and B, used for link relations R_A and R_B respectively, the useable buffer set up is m pages, A buffer as m_1 pages and B buffer as m_2 pages. The main idea of our link algorithm is to scan R_A only once, but scan $R_B \left\lceil \frac{P_A}{m_1} \right\rceil$ times, where P_A is the number of disk pages taken up by the R_A element group. m'_1 is the number of pages occupied on disk by the R_A element group placed in the buffer. This is because the relational element groups on the disk do not exceed the page load therefore $m' \geq m_1$. The input cost of the link operation according to this algorithm is:

$$CIN(Q) = P_A + \left\lceil \frac{P_A}{m_1} \right\rceil \cdot P_B$$

in which P_B is the number of pages of disk occupied by the R_B element group. Obviously, the optimum buffer allocation method is:

$$\begin{cases} m_1 = m-1 \\ m_2 = 1 \end{cases}$$

The optimized processing used in a single element query is also used in a two element query.

IV. MRDS Database Manager Subsystem

1. Introduction to functions

The MRDS database administrator subsystem is a local single user database management system designed for database administrators (DBA): it sets up a database for the DBA and makes it convenient for managing data locally. This subsystem includes an all-net online query function and provides methods for generating a database and managing data, that is, the database administrator subsystem has all the functions of the relational query language SRQL. The system structure is illustrated in Figure 6.

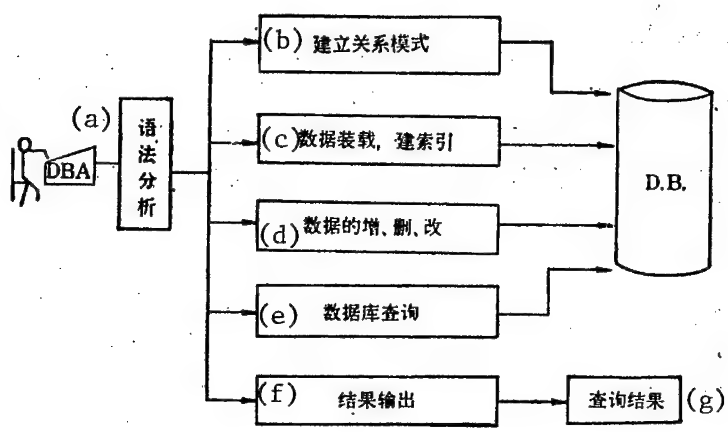


Figure 6. Diagram of MRDS Database Manager Subsystem

Key:
- a. syntax analysis
- b. establish relation mode
- c. data load, set up index
- d. add, delete, revise data
- e. database query
- f. result output
- g. query result

Database generation is an important task of DBA. This subsystem provides statements for creating modules, statements for loading data, statements for deleting modules, and statements for displaying relational modules. After the DBA abstracts the real world into relational modules, the statement for creating modules can be used to set up all the relational modules in the database and the load data statement be used to load the managed data into the database. The display relational modules statement and delete module statement provide the means for the DBA to revise and restructure database modules.

Manipulative functions for adding to, deleting, and revising data are provided by the insert statement and delete statement in the subsystem. Data added, deleted and revised can be the view resulting from a query as well as data input from the control console used by the DBA.

The subsystem also provides all the functions for local execution of an online query and the DBA can carry out queries and output the results of queries of the database for local users.

In the database administrator subsystem, the databases established and maintained by the DBA can be opened to online queries of terminal users on the RDC Net at DBA option.

The online query method has already been discussed above, and will not be repeated here, but the method of implementing database generation and data manipulation will be discussed.

2. Database generation

Database generation is made up of two parts: the creation of database and the loading of the data.

On the basis of the relational names and attribute specifications given by the creation module (including attribute name, class, and precision), the system estimates the maximum number of element groups and the index specifications, generates the relational module and index information and creates the relational files and the index files of attributes which must be created.

The relational module is the central data structure of this system. It reflects the structural information of the database and is the major basis of the query, load, add, delete, revise and output operations. The relational module takes up one page (512 bytes) and is made up of the relational description and the description of certain attributes and is stored in the first page of the relational file (as illustrated in Figure 7).

relation description
attribute 1 description
⋮
attribute n description

Figure 7. Relational Pattern Structure

pattern page
function page
element group space
character space

Figure 8. Relational File Organization

Relational files and index files are set up when the module is created and are managed by the RDOS file system. The relational files are continuous files. When the module is created, a continuous disk area is requested on the basis of the estimated maximum number of element groups and the average length of the pointer attributes. The structure of relational files is uniform, and is made up of module page, function page, element group space and character space (Figure 8). Character space stores the character strings of the pointer class data. When the relation has no pointer attributes, the character space length is 0.

Continuous files cannot change their size dynamically, thus, the estimate of the maximum element group number is a parameter that the DBA cannot overlook.

Index files are random for speed of search and ease of maintenance. The system automatically adds the attribute name of the index to be established after the relational name to form the index file name. Each relation must determine an attribute as primary key and create an index on it.

The production of the database is completed by loading data in existing relational pattern. The system provides many load paths and the DBA can use the console and system interaction to load data and can also batch load data from disk files previously created at a terminal. Whatever load path is used, the data is written in a uniform external format and from the user's perspective, this external format is easy to understand.

In the data loading process, each element group loaded inserts a corresponding index page into the index file on the basis of the relational pattern.

Relational logic space can be divided into two parts. One part is continuous empty areas made up of areas not yet used since the relation was set up. The other part is space released on the link and is space created by deletion of element groups. When loading, this system first loads empty areas and only when the empty areas are full does it search for a released link. Since attribute length is irregular, the released link search technique is not used for pointer type attribute character space. After the empty areas are full, the recombination character space method is used for determining the useability of character space.

3. Data manipulation

Adding, deleting and changing data manipulation capabilities are indispensible to a database and the insertion and deletion statements in this subsystem provide convenient data manipulation functions.

When a certain element group or class of element group must be deleted from a database, first of all the set of element group to be deleted must be defined. This subsystem conveniently lets the DBA define the set of element groups to be deleted according to any attribute. The view set up using a query can be deleted from a relation at one stroke. When the set of element groups to be deleted is controlled by the primary keyword, the system provides a function allowing the DBA to delete element groups one by one using the console.

There are also two paths for the element group insertion process, i.e., the element groups can be inserted by the DBA using the console and relations can also be inserted through the view set up by the query statement, but at such times, between the relation and view inserted, the number and class of attributes must be the same.

V. Implementation of Index Operations and Index File Management

MRDS index files are created with the support of the RDOS file management system and index files managed by the index unit (IU), query statements executed by SRQL, and services provided in the data load and operation statement process are illustrated in Figure 9.

Through a correspondence of the key work recorded (i.e., element groups) and the logic number (element group number) recorded, and on the basis of the relation of the element group sequence number and the element group disk sector address, the IU creates a correspondence between the key word and the element group recorded and provides the database system with an entry point type access mechanism.

In carrying out the adaptive screen operation of the query statement, the IU divides the index files by operator and creates a result element group sequence number set in preparation for the set operation of the adaptive select. This avoids scanning element group space directly and raises query efficiency.

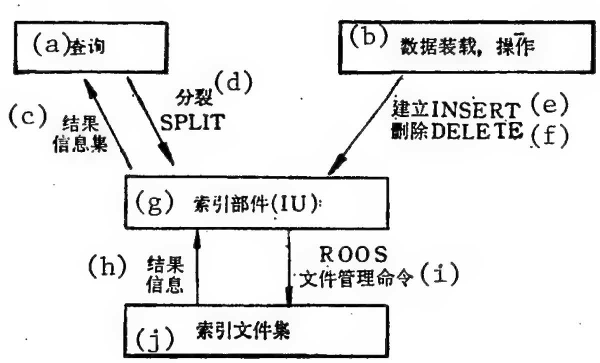


Figure 9. Index File Management

Key:

- a. query
- b. data load, operation
- c. result information
- d. split SPLIT
- e. set up INSERT
- f. delete DELETE
- g. index unit (IU)
- h. result information
- i. ROOS file management command
- j. index files

In executing the SRQL database load and operations statements, the IU is responsible for inserting, deleting, and updating the index pages of corresponding index files and carrying out operations of related indexes according to the index statements.

In addition, the IU is also responsible for organizing and managing index file disk sectors.

1. Logic structure of index files

In MRDS, relational element groups are stored in element group space according to LAST sequence. The system gives a sequence number according to this sequence. The index set up on the primary key work attribute of the relation is the prime index and the one on the secondary key word attribute is the secondary index or inverted index. A relation must have a primary index. An index file for an attribute is created for a corresponding attribute index.

An index is made up of an index head and an index body: the index head stores the parameters of the index and the index body is the set of index pages and the index item consists of the key word value and the corresponding element group sequence number.

2. Creation of an index and the data operations in the index

First of all, consider the following:

There are the two relations DEPT (DNO, DNAME, LOC)

EMP (NAME, DNO, BYEAR)

Now suppose we want to create a primary index on the relational DEPT attribute DNO and the NAME of DNO, and create an inverted index on the DNO and BYEAR of EMP. The DBA will have to input the following statements:

(1) CREATR DEPT (DNO:INTEGER(3),DNAME:STRING(10),LOC:STRING(5))100 INDEX KEY DNO;

(2) CREATR EMP (CNAME:STRING(10),DNO:INTEGER(3),BYEAR:INTEGER(4))100 INDEX KEY NAME,DNO,BYEAR;

When executing these two statements, the system sets up a random file for each attribute and prepares to receive element groups. When the element groups are loaded, the system scans each component in sequence and when it encounters a component which corresponds to the index set up, then using this component as key word it sets up an index page with this word and the element group serial number and the IU defines the disk sector where this index item should be stored and stores this index item and carries out the necessary disk sector management. The results of the IU operation on the above statement is illustrated in Figure 10.

As far as operations on indexes are concerned, insertions are the same as creating an index. When deleting, the item to be deleted must be found and the disk sector must be reorganized. Revising an index is implemented by a combination of deleting and inserting.

Relation EMP

Element group serial no	NAME	DNO	BYEAR
1	ZHAO	1	1957
2	QIAN	2	1958
3	SUN	3	1957
4	LI	3	1958
5	ZHOU	2	1959
6	WU	1	1958
7	ZHENG	2	1957
8	WANG	1	1959

Primary index

NAME	ZHAO	1
	QIAN	2
	SUN	3
	LI	4
	ZHOU	5
	WU	6
	ZHENG	7
	WANG	8

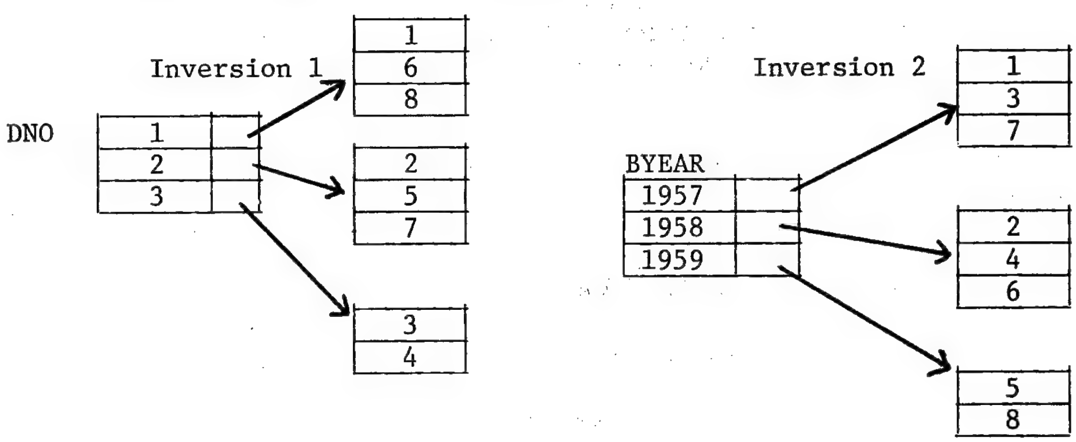


Figure 10. Index File Example

3. Processing character string indexes

The following situations are encountered in data processing and information searches: when there is more than one AUTHOR to an article, multiple author name keywords correspond to one PAPER element group. At such times, the AUTHOR attribute index is called a character string. The IU provides a function to process a string index.

The general form of the string is: $ST = S_1\&S_2\&...\&S_k$

where S_i is any character string not containing an &.

When setting up a character string index, the IU recognizes the conjunction '&', fetches each character string and processes it with the many character string index pages which match the element group serial number.

4. Organization and management of index files

On the basis of MRDS hardware and software environment, the following forms and methods are adopted for index file organization and management:

(1) The random file provided by RDOS is the basic organizational form of the index files, an attribute index file acts as an RDOS random file. In this way, the system file management commands provided by RDOS can be used to call and manage index files and take into account the two advantages of random files: flexible expansion of disk sectors and rapid disk access.

(2) Carrying out disk positioning by zacou [7177 0410] algorithm. In this way it can match better with RDOS random files and rapidly define disk sector speed.

(3) Organizing index file disk sectors by the tree chain combination mode using unused disk sectors which already exist in random files as much as possible. The three chains are: ① Basic sector chain: the disk sector on this chain is those disk sectors defined after going through a zacou operation. ② Overflow sector chain. This is a chain organized of disk sectors requested when zacou results "impinge". ③ Empty sector chain: when index files must request a new overflow sector, the empty sector chain is checked and first priority use of this chain's empty sectors becomes new overflow sectors.

(4) Clear garbage, i.e., to reclaim unused disk sectors for the system, IU is implemented through rewrite file. IU first rewrites nonempty disk sectors in index files to an intermediate file, reserves the index file name, deletes the index file, and finally changes the intermediate file name to the original file name.

MRDS functions have been implemented on the DJS-100 series computers. The two subsystems have a total of over 40 K of assembly programs and since 1981 over 5 man-years have been expended.

Implementation of multiple user queries in a network environment provides favorable conditions for further research and implementation of a distributed database.

8226/12712
CSO: 4008/244

REMOTE JOB ENTRY SYSTEM DESCRIBED

Beijing JISUANJI XUEBAO [CHINESE JOURNAL OF COMPUTERS] in Chinese Vol 7 No 1, Jan 1984 pp 59-70

[Article by Zhang Yaolin [1728 5069 2651] Graduate School, Chinese Academy of Sciences, Lin Chuang [2651 7068] Computer Center, State Planning Commission, Gao Yuchang [7559 3254] Institute of Applied Mathematics, Hebei Province Academy of Sciences]

[Text] I. Introduction

In the RDC Net, the three protocol mechanisms below the transport level provide transmission service for process communication, resolving data transmission between the processes or devices of host computers linked to the network. We designed a service level protocol--the remote job entry protocol--to manage and use network resources. In addition, on the basis of the current situation and characteristics of the RDC Net, a Remote Job Entry (RJE) System, or RJE System, for short, was implemented on the DJS-100 series computers.

Under the support of the RJE system, from the terminal, the user can send the service host computer a job, query the execution situation, send job results to the network and communicate with other users, and thus share the hardware resources of the host computers on the network.

The RJE system expanded the original batch processing system of the operating system, increased the job scheduling function, expanded the job flow input mode, and improved the job management mechanism.

II. Foundation and Design Principles for Developing the RJE System

The link level, packet level, and transport level protocol mechanisms of RDC Net communications software are the foundation of higher level protocols. The RJE system implements communication between the end user and the RJE service mechanism program through the communications software.

To resolve problems of the different characteristics of the terminals linked to the network, the packet level assembly/disassembly program (PAD) manages meeting, connect/disconnect of the link, code conversion and data transmission between the user terminal and the service program. Since the internal

capacity of the DJS-100 series computer is limited, the RJE protocol mechanism replaces the user program with the PAD, which will be discussed in detail below.

The RDC Net file transmission protocol (FTP) provides file transmission service between different host computers.

The RDOS operating system of the DJS-100 series computers has a batch processing system and a batch command, which is similar to the job control language of keyboard command language. Although the original system's batch command function is weaker, the original batch command was not altered in order to concentrate the primary energies on implementing remote job entry and job management.

Under the support of the above software resources, we designed the RJE protocol, implemented the RJE service mechanism, monitor program and account program. Our design principles are:

•change the original RDOS as little as possible;

•make it easy and flexible for the user;

•easy to expand by appropriate later developments.

III. Overview of RJE System

1. Position in the network

The RJE protocol is a protocol directed to the applications programs and is one of the three basic service level protocols. It is situated at a high level in the network hierarchy. Figure 1 shows the logic structure of the RDC Net in which the low level protocols, i.e., link level, packet level, and transport level protocols, provide transmission service to the higher level protocols.

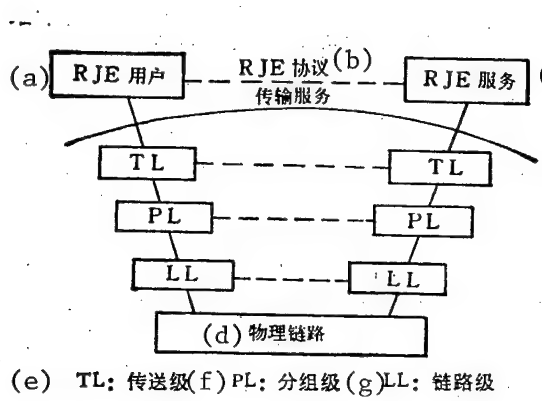


Figure 1. Structure of RDC Net protocol levels

[Key on following page]

Key:

a. RJE user
b. RJE protocol transmission service
c. RJE service
d. physical link
e. TL: transport level
f. PL: packet level
g. LL: link level

2. Component parts of the RJE system

The RJE system is made up of three parts, namely, the RJE protocol mechanism, monitor and control program, and the account program.

Under normal circumstances, the service level protocol mechanism is composed of two groups of processes: the user process and the service process. The former converts the user's local commands into protocol commands and converts the responses to the protocol commands into local responses. The latter interprets the protocol execution commands and provides the service. In the RJE system we have implemented, local commands and protocol commands are considered identical and to reduce the internal load of the user process on the host computer, the user process is not installed. Part of its work (such as setting up the link with the service process) is completed by PAD. For this reason, the RJE protocol mechanism is also the service mechanism. The RJE service mechanism is the main body of the system: it expands the service host computer to many virtual computers and at the same time can establish links with many user terminals, interpret protocol execution commands, and provide remote job entry.

There are two aspects to the main task of the monitor and control program: one is controlling and monitoring the operation of the RJE system, including installing the network environment, starting and stopping RJE service, and reporting on system operating states; the other is managing jobs, including managing job queueing, scheduling job operations, and controlling execution of the Batch subsystem. Since the DJS-100 series computer acts as both node computer and resource host computer in the network, its operating system cannot support the simultaneous operation of several subsystems, thus when the Batch subsystem is working, activity on the network must stop. Since job entry and job execution cannot be carried out simultaneously, it can only operate in stages. These two stages are called the entry stage and the execution stage, respectively. The transfer of stages is carried out by the monitor and control program.

Using the system under a network environment is different from using a single computer system. The administrator cannot directly supervise the user and determine whether or not the user can use the system. For this reason, the account program was designed. The account program is a service program under the RJE system. It manages user account files, terminal account files, and user accounting. It is an independent program and is also called the account system.

3. Network environment of the RJE system

Above we explained the relations of the RJE system and the network and their internal structure. Below we will explain the operating environment of the RJE system and its functions using the example of a network composed of three node computers.

As illustrated in Figure 2, the host computer A is operating an RJE system. The user at the terminal is connected to host computer B, and host computers B and C operate other service systems. Suppose host computer A already has job specs and a source program file and the user on the terminal wants to submit data files and a job to the service host computer A. After being carried out on host computer A, the results are sent to host computer C.

IV. Interface with User Personnel

User personnel includes end users who have requested system service, service host computer operators who maintain and manage system operation, and security personnel who manage user accounts and collect fees. Their interfaces with the RJE system are the RJE protocol commands (also called user commands), operator commands, and account commands, respectively.

1. RJE protocol

RJE protocol refers to the rules for the user to employ remote batch processing on the network. Currently, there is no standard text which can be observed. The standard text of the protocol also differs on different networks. However, it must include the following basic functions:

•submit/cancel job, including submitting job specs, source program and data files;

•query a job execution situation;

•fetch job results or send results to network.

The protocol we designed includes the above basic functions, as well as the auxiliary functions of setting up communications between users, querying files and deleting files. Since entry and execution are carried out in stages, we did not create a command for initiating the execution of batch processing.

For the convenience of the user, the protocol commands and responses are obvious to the user. The protocol adopts a command language in interactive dialogue form and the right to speak is determined by the system. For this reason, the user must obtain a command prompt (R), message prompt (I), or telegram prompt (?) from the system before a command, a message, or an interrupt can be sent. The command response is a text message whose meaning is clear so that the user can readily understand the execution state or results of the command.

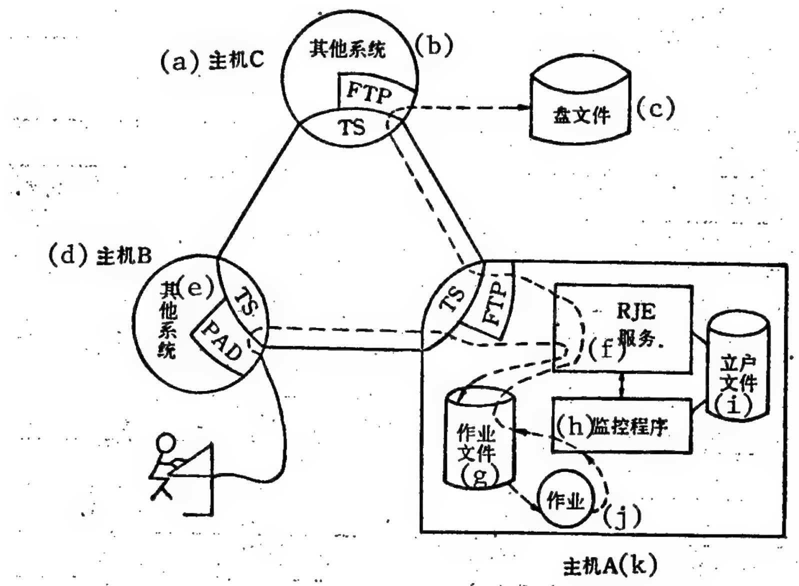


Figure 2. Remote user using RJE system

Key:

a. host computer C
b. other systems
c. disk files
d. host computer B
e. other systems
f. RJE service
g. job files
h. monitor and control program
i. account files
j. job
k. host computer A

There are 11 commands in the protocol. Each command is made up of the name of the command and the relevant parameters concluded by a carriage return ("CR"). Table 1 lists all the commands and their corresponding functions.

The protocol has two kinds of interrupt so that the user can control the file input/output process in realtime. These interrupt messages act as telegrams sent to the system.

2. Operator commands

Operator commands are techniques for the service host computer operators to supervise and control system operation. Ten commands were created focused on how to manage the RJE system. Corresponding commands were not created for tasks which could be carried out using the keyboard commands of the original

operating system thereby reducing unnecessary redundancy. The operator command language also uses the interactive dialogue mode. Its primary functions are:

•start/stop RJE system operation;

•start/stop network activity;

•start batch processing, schedule job execution;

•query system work status;

•communicate with user.

Table 1. Table of User Commands

Class	Command name	Function
Meeting control	RJON	First command after user sets up link, supplies user name, password and terminal
	RJOF	Requests turning off link between RJE service and terminal, concludes dialogue
File input output management	PATH	Sets up path for accessing files on service host computer
	IPUT	Input a file to the service host computer from indicated host computer or user's terminal
	OPUT	Output a file from service host computer to indicated host computer or user's terminal, carry out output of result files
	ENQF	Query length, characteristics, and creation time of indicated file on service host computer
	DFIL	Delete indicated file on indicated host computer
Job management	JOB	Submit job, indicate job specs, disk file name and system output file name
	ENQ	Query indicated user's job situation, i.e., number, and time of job submission and sequential number in the queue
	DJOB	Delete indicated job
Message exchange	SEND	Send information to host computer operator, terminal operator or other user

Table 2 sets out the operator commands and their functions. Each command is made up of the command name and its parameters and is concluded by a carriage return ("CR").

Figure 3 depicts the transfer of stages during system operation. The commands that initiate changes in state are indicated on the arcs.

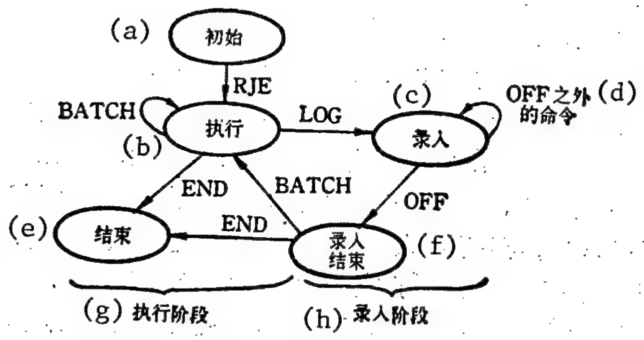


Figure 3. State diagram of system operation

Key:

- a. start
- b. execute
- c. entry
- d. command outside of OFF
- e. end
- f. entry end
- g. execute stage
- h. entry stage

Table 2. Table of operator commands

Class	Command name	Function
System process control	RJE	start RJE system operation
	END	end RJE system operation
	LOG	begin entry stage
	OFF	terminate entry stage
Job management	BATCH	schedule execution of user job
	DELETE	delete indicated job
	COPY	transmit files to host computer
Message exchange	EMIT	send message to user
	GET	get message sent by user
Query	STATUS	query system state and job situation

3. Account commands

Account commands are for the use of security personnel and interact with the account program to alter and query user account files, terminal files, and user accounting files. Accounts are set up on the system for users and terminals.

The user must set up an account on the system through the security personnel before he can obtain the right to use the system. On the other hand, the system judges whether or not a user is legitimate on the basis of the contents of the account files.

The account commands provide the functions of adding new accounts, changing user name and password, tabulating output user information and user accounting. All the account commands and their functions are tabulated below.

Table 3. Table of account commands

Command name	Function
ACCOUNT	starts account program operation
ADD	open account for new user and new terminal
CHANGE	change user name and password and account terminal name
DELETE	delete existing user and terminal account
LIST	list and output messages of users and terminals with accounts and user accounting
END	end account program operation

V. Implementation of RJE System

The RJE system monitor and control program and service mechanism is made up of two groups of processes. On the basis of the principle of not changing the original operating system, these processes function as RDOS user processes and are scheduled by the RDOS task dispatcher. Since the account program interacts through account files and with two other parts, it has a certain independence. For this reason, we will first deal with the implementation of the monitor and control program and the service mechanism.

1. Monitor and control program

Earlier we mentioned that the important functions of the monitor and control program were to monitor and control system operation, manage the job queue and schedule job operation. These functions are implemented by the operator command interpreter and the job management program.

1.1 Monitor and control processes

The monitor and control process is a root task of the RJE system which is set up as soon as the system is initiated and operates the operator command interpreter. After the process is started, it waits for an operator command. After a complete command is received, the command is analyzed according to the command name table, the command syntax is checked, and then it is transferred to the relevant command execution program for execution. The relevant response message is given on the basis of analysis or the results of execution. Finally, it is again placed in a command waiting state.

The monitor and control process interacts only with the RJE service and FTP mechanism. It does not directly use the transmission service of the network but transmits files and sends messages on the network through the FTP mechanism.

When the entry stage begins (receiving a LOG command), the monitor and control process sets up an RJE service management process and at the same time sets up a network environment. When the entry stage is concluded (receiving an OFF command), it registers a STOP processing unit with the RJE service management process. After RJE service has stopped service, it notifies the monitor and control process and then cancels the transmission service processes.

When operators transmit files and send information to other host computers through the COPY and EMIT commands, the monitor and control process files a request unit in the transmission request area of the corresponding FTP process and wakes up the process. The request unit includes remote host computer name, file transmission direction, source file name, and target file name and structure. After file transmission has concluded, the FTP process notifies the monitor and control program of the results of the file transmission.

In the execution stage, on the basis of the parameters of the BATCH command, the monitor and control program calls the job scheduling module and selects the job to go in this line.

1.2 Job management program

The job management program implements the job management of the RJE system, and expanding the batch processing functions of the original operating system. Depending on the differences in functions, two modules were designed, one for job queueing management and one for job scheduling.

1.2.1 Job queueing management module

This module is constructed of the job queue and one of the operations defined in this important data structure. These operations include inserting jobs, deleting jobs, and searching for jobs by user name and job name. The important functions of the module are:

•storing jobs submitted by the user through RJE service according to entry sequence;

•querying jobs, including job execution state, serial number in the queue, and submission time;

•deleting indicated jobs;

•accessing an indicated job from the queue.

As illustrated in Figure 4, job queue is made up of the job queue control table, the job directory queue and the job control block files. For saving internal space and for ease of searching, when the RJE system is operating only the first two parts are resident internally.

The job control block files are continuous files which can store 99 job control blocks. Each control block is made up of 32 bytes and records the job name, system output file name, and time of submission (day, hour, minute).

The maximum length of the job directory queue is 99 records. Each record takes up only one byte and records the log number of the job submitter. When the record item is 0 it means that it is idle and can be used or that the corresponding job has already been deleted. The directory and the job control block are in one-to-one correspondence and the file pointer of the corresponding file control block can be calculated from its relative position in the directory queue, e.g.:

File initial pointer + 64*(directory address - first address of directory queue)

thus by resetting the file pointer, the corresponding control block can be found at once.

The queue control table records the current state of the queue.

The queue control management module is called by the monitor and control process and the RJE service processes, thus it uses the P and V operations to implement exclusivity.

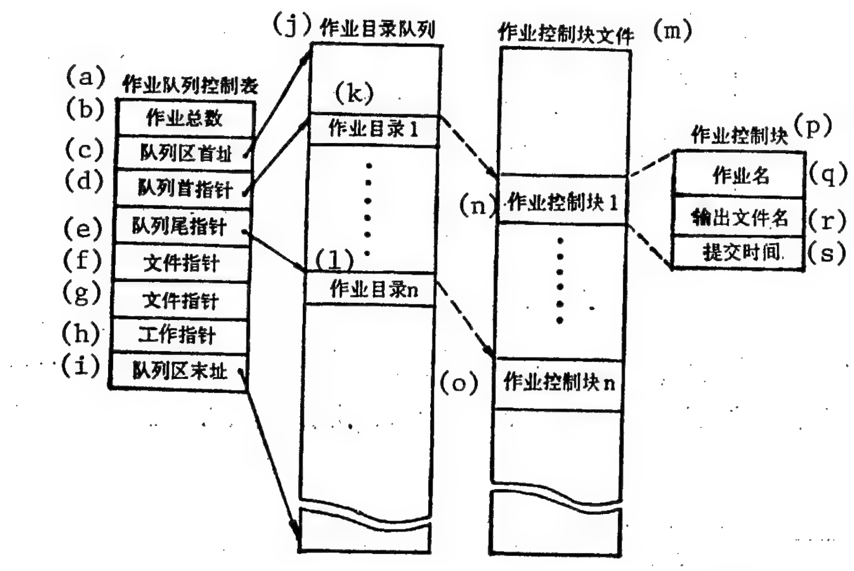


Figure 4. RJE system job queue

Key:

a. job queue control table
b. job total

[Key continued on following page]

c. first queue sector address
d. first queue pointer
e. queue end pointer
f. file pointer
g. file pointer
h. work pointer
i. last queue sector address
j. job directory table
k. job directory 1
l. job directory n
m. job control block file
n. job control block 1
o. job control block n
p. job control block
q. job name
r. output file name
s. submission time

1.2.2 Operations schedule module

This module is the module that executes the stage operations. It calls the queue management module according to the scheduling principles and accesses the job control block from the job queue, creates the RDOS keyboard command file, and then transfers it to the Batch subsystem. The Batch subsystem interprets the batch command in the job execution specs and executes the job. The job execution process is recorded in the system output file and the accounting file records the job operation time. After the job execution is completed, the Batch subsystem passes the RJE system back to the internal storage, and goes on to schedule the next job or waits for a new operator command.

The job schedule principle is very simple. It uses a combination of the system and the operator, i.e., the system selects the job on the basis of the principle of first come, first executed, and permits the operator to indicate which job to carry out.

2. RJE service mechanism

The RJE service mechanism is the main body of the RJE system: it is under the support of the communications software and FTP, is set up on the RDOS foundation and is controlled and operated by the monitor and control program.

2.1 Interface with other mechanisms

Figure 5 describes the relations between RJE service and the monitor and control program, FTP mechanism, transport stations (TS), and PAD.

The interface with the monitor and control program has already been introduced in the discussion above concerning the monitor and control process.

The interface with FTP is similar to the monitor and control process.

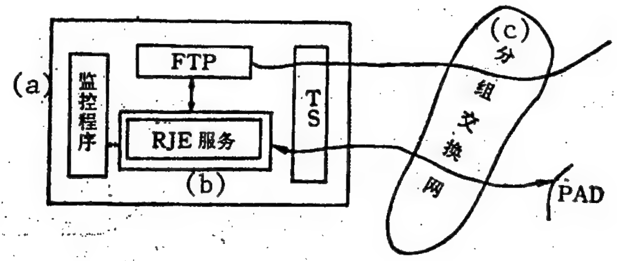


Figure 5. Interface of RJE service and other mechanisms

Key:
a. monitor and control program
b. RJE service
c. packet exchange net

From Figure 2 it can be seen that dialogue between RJE service and the user is implemented by TS. In the structure of our system, even if the user's service host computer is the local host computer, it is no exception. The link of RJE service and TS adopts the mode of cross registration of processing units. The content of the processing unit is the primitives supplied by TS.

PAD replaces the user process to save memory. However, PAD is not concerned with the meaning of user commands and interrupts, but only organizes them into letters and telegrams and sends them to the network. Figure 6(a) and 6(c) point out the letter and telegram format that PAD sends to RJE service. Since RJE service and the user employ interactive dialogue and RJE service has jurisdiction over determining the right to speak, the letters sent by PAD are a complete user command or a message, at the same time, sending a letter means handing over the right to speak. RJE service also adopts the mode of notifying PAD of response command execution and interrupt processing situation. Letter format is as illustrated in Figure 6(b). The difference with Figure 6(a) is whether or not the marker M giving the user the right to speak is added at the end of the letter.

2.2 Internal structure of RJE service

RJE service is made up of a group of processes with different levels of priority. This group of processes includes a management process (MP), a user interrupt process (UIP), a timing process (TP), and certain dialogue processes (DP). These processes are produced simultaneously when the entry process begins.

Certain dialogue processes are subprocesses of the management process and serve as RJE service resources managed by the management process. Each dialogue process has a process level software clock and these clocks are uniformly managed by a timing process. The user interrupt process is responsible for accepting an interrupt sent in telegram form by the user and converting it into internal code and sending it to the corresponding dialogue process. RJE service structure is shown in Figure 7. Arrows indicate the links between processes.

(a) PAD to RJE service letter format

Key: a. text information

(b) RJE service to PAD letter format

Key: a. text information

(c) Telegram format

Key: b. message

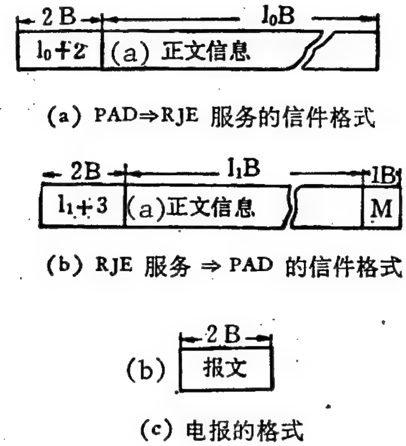


Figure 6. Letter and telegram formats exchanged between RJE service and PAD

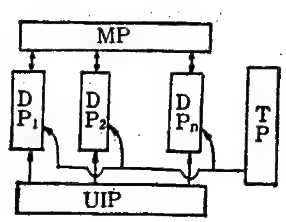


Figure 7. Internal structure of RJE service

2.3 Functions and implementation of RJE service processes

2.3.1 Management process

Management process is set up by the monitor and control program. The interface to process and monitor and control dialogue when service is being provided sets up the link with the user requesting service and manages the dialogue process. The specific implementation is as below:

•listening for user requests on the network.

The management process has a network-wide unified 'information port' [xinkou [0207 0656]]. When service is being provided, the information port is opened to listen for user requests and in a passive link state.

•scheduling dialogue processes, receiving users.

The dialogue process set up by the management process begins in the idle state. Once it hears a user request, the management process assigns an idle dialogue process for this user service, cuts to a link, registers a START processing unit to the selected dialogue process so that it is in the busy state. If there are no idle dialogue processes, the management process rejects the user and breaks off the connection. After processing a user request it goes back to listening.

•terminating service

After a monitor and control process STOP processing unit is received, the management process stops listening, turns off the information port and notifies the dialogue process to conclude the current task. After all dialogue processes have stopped, the management process cancels the dialogue process, user interrupt process and the close process. Then it notifies the monitor and control process.

Figure 8 describes the work flow of the management process.

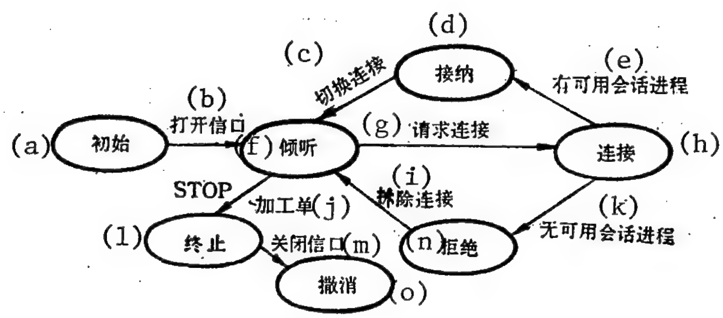


Figure 8. Flow process of management process

Key:

- a. start
- b. open information port
- c. switch link
- d. accept
- e. useable dialogue process
- f. listen
- g. request link
- h. link
- i. processing unit
- j. disconnect link
- k. no useable dialogue process
- l. terminate
- m. close information port
- n. reject
- o. cancel

2.3.2 Dialogue process

Dialogue process is the nucleus of RJE service. The main functions are dialogue with the user, accepting letters from PAD, interpreting execution protocol commands, and sending letters to PAD explaining command execution.

Each dialogue process has an information port which is opened during the service period. When a START processing unit is received, a link in the data transmission state is already established. After the dialogue with the user is concluded, the disconnected link simultaneously reverts to its idle state.

Many dialogue processes are set up in RJE service and execute the same public module thus expanding the service host computer to many virtual computers serving users simultaneously. From the user's point of view, once he is accepted, there is an interactive dialogue service host computer to serve him.

The public module which executes the dialogue process is made up of the following submodules:

•receiving module. It receives letters from the network including reserve buffers and responds to letters received from TS.

•command analysis module. It analyses the letters received by the editor to determine whether they are commands or messages, checks the legality of the command, and relays it to the relevant command processing module.

•command processing module. There is a processing module for each protocol command. It checks the syntax of the command and loads the response into a letter depending on the work the lexeme is to complete.

•transmit module. It sends the loaded letter to the other end of the link.

The public module can be reentered and thus each process has an independent work sector and data sector. Figure 9 illustrates the data structure on which the dialogue process relies.

In it, the parameter sector, message sector, buffer sector, clock element, FTP request unit sector and online terminal name table address state is assigned to each dialogue process.

The path name table is managed by the PATH command processing module. Each dialogue process can at most occupy two table items, i.e., two paths at most can be established. Each table item includes the path name and the number of users. When the user dialogue is concluded, the number of users of the established path is decremented by 1, and if the result is 0 then the path is closed.

The clock table is managed by the timing process.

To resolve the problem of contention when several users simultaneously access a disk file or use the same output device, the following strategy has been adopted. A disk file can be read by several users simultaneously, but only one user at a time is allowed to revise it. Revision rights are assigned according to the order of user requests. Users who do not have the revision rights temporarily must wait until they receive revision rights on repeated requests before the corresponding command can be executed. The same principle of one user at a time also must be observed with regard to the output devices. The difference between them and the disk file processing is that they need not submit repeated requests, because exclusivity is implemented by the P and V operations.

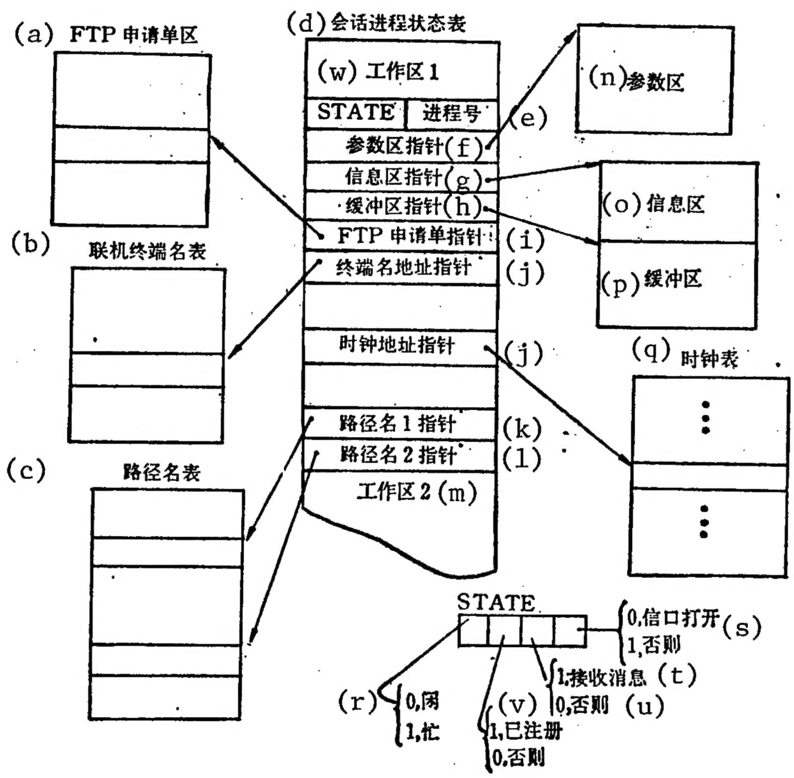


Figure 9. Table of relations with dialogue process

Key:

a. FTP request sector
b. online terminal name table
c. path name table
d. dialogue process state table
e. process number
f. parameter sector pointer
g. message sector pointer
h. buffer sector pointer
i. FTP request unit pointer
j. clock address pointer
j. terminal name address pointer
k. path name 1 pointer
l. path name 2 pointer
m. work sector 2
n. parameter sector
o. message sector
p. buffer sector
q. clock table
r. 0, idle, 1, busy
s. 0, information port open, 1, or else
t. 1, receive message
u. 0, else
v. 1, logged, 0, else
w. work sector 1

To implement communications between users, a message letter is set up corresponding to each terminal which records the messages sent to the account of the user at that terminal. When the terminal operator requests service, the contents of the message letter is output on the terminal and sent to the user by the operator.

2.3.4 User interrupt process

The priority of the user interrupt process is higher than that of the dialogue process and the user interrupt process is responsible for receiving telegrams on the network. It converts the telegrams into internal code and notifies the corresponding dialogue process.

2.3.5 Timing process

The timing process is a fixed time task. It manages the software clock of each dialogue process and starts once for each unit of time.

To improve the use rate of resources, when the dialogue process is conducting an interactive dialogue with the user, it demands the user must respond within five minutes after he receives the prompt (by entering a command, a message, or an interrupt character). Otherwise it believes that the user has abandonned use of RJE service.

On the other hand, when the dialogue process sends a message to the user, if a response successfully sent by TS is not received within a certain time due to the circuit or other problems, it believes that the link cannot be used. Thus it stops user service of this link.

To implement this function, when the dialogue process is sending a letter or waiting for a letter (or telegram), a waiting time value is entered in its software clock. When the timing process is working, each software clock in the clock table is checked. The value of non-0 value software clocks is decremented by 1. If the result is 0, then the out-of-time processing unit notifies the corresponding dialogue process. After all software clocks have been checked, it is again in on-hold state.

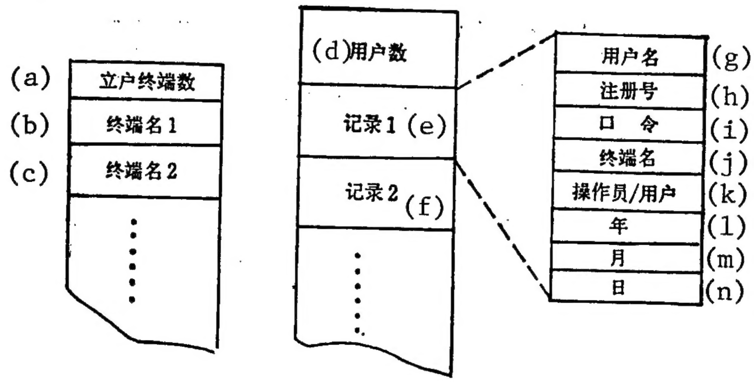
2. Account program

The account program manages terminal account files, user account files, and user accounting files and interprets execution of security personnel account commands. The account program operates on the network independently. The emphasis in implementation is in determining the structure of the above three files and providing search, insert, revise and delete file contents operations, and in selecting suitable technology to guarantee the security of the system.

Terminal user files record the terminal name and terminal total number of service host computer accounts. The system sets up corresponding messages accordingly.

User files record the time when the user employed RJE service and when he stopped and the time used to execute a job.

There were two considerations regarding system security. One prohibits illegal users from using the system and the other prevents non-security personnel who use the account system from destroying account files and accounting files.



*(a) 终端立户文件 *(b) 用户立户文件

Figure 10. Account program management files

Key:

*(a) terminal account file
*(b) user account file
a. number of account terminals
b. terminal name 1
c. terminal name 2
d. number of users
e. record 1
f. record 2
g. user name
h. log number
i. password
j. terminal name
k. operator/user
l. year
m. month
n. day

To prevent illegal users from using the system, in addition to checking the user name, the system also demands that the user provide his password. The password is determined when the user account is set up, and is converted by the account program into a secret code that is stored in the user account files. This saves on the space used by the account files and reduces the possibility of break in.

Since the security of account files is more important than prohibiting illegal users, the account program has a response type password which checks the user's status. That is, when the account system is initiated, the system issues a random number and the user is issued a corresponding password which is a number based on an established conversion coefficient.

VI. Conclusion

The RDC Net RJE system has been implemented on the DJS-100 series computer and has the characteristics of an online system with the functions of a distributed network service system. The remote user can use the software and hardware resources of service host computers conveniently and flexibly at terminals on the network.

The RJE system design took into account future developments and can be expanded and new protocol commands can be added easily.

We experimented with our work by implementing the service system aspects on a minicomputer network and we are exploring experiments to further develop network resources.

8226/12712
CSO: 4008/244

END